

"الخطوة الأولى نحو البرمجة للويب"



م. مُختار سيِّد صالح

تصميم الغلاف للمصمّم المبدع :

بدر عبدالعزيز الإبراهيم

تعلَّم

HTML5 & CSS3

(الخطوة الأولى نحو البرمجة للويب)

المهندس مختار سيِّد صالح

الإهداء

إلى أعلى نخلتين على ضفاف القلبِ ...

والديَّ حفظهما الله.

تنويه

فرغتُ من تأليف هذا الكتاب في الأشهر الأولى من عام ٢٠١١ ، و كنت قد عقدتُ العزمَ على إصداره مطبوعاً كأوَّل كتاب عربيٍّ في موضوعه ، حاذياً بذلك حذو أخيه و سابقه (تعلّم jQuery في ١٢٠ دقيقة) ، لكنَّ مشيئة الله قضت ألّا تتهيّأ الظروف الملائمة لذلك حتّى هذه اللحظة بعد انقضاء خمسة و عشرين شهراً على إتمامه ، و لأنَّني أؤمن أنَّ قيمة عملٍ كهذا تتضاءل مع مرور الوقت فقد رأيت –بعد تفكير– أن أنشره مخلصاً بشكلٍ مجّاني بصيغة الكتاب الإلكترونيِّ (PDF) لعلَّ أحد الدّارسين ينتفع به لمّا كان نشره بهذه الصّيغة لا يتطلّب منّي أكثر من كتابة هذه الأسطر.

مختار

البوكمال ٢٠١٣/٦/١٠

بين يدي الكتاب

بسم الله الرّحمن الرّحيم ، و الصَّلاة و السَّلام على نبيِّنا محمَّدٍ و على آلهِ و أصحابهِ أجمعين، و بعد:

لم يَعُد الـ Web اليوم مقتصراً على تقديم المعلومة بشكل نصِّي بسيط، حيث أصبح من المهم تقديم المحتوى بشكل و أسلوب أقرب ما يكون إلى الإعلامي، بل ربَّما هو كذلك فعلاً، فقد أصبح من النادر -برأيي- أن تجد في مواقع الـ Web اليوم -و أعني المواقع الجيدة منها- ما لا يقدم المعلومات بشكل مسموع و مرئي إضافة إلى الشكل التقليدي المقروء و هذا من ناحية، أما من الناحية الأخرى فقد تطوَّر الـ Web ذاته كفهوم مع بدء التنظير لما يعرف بالـ Web 2.0 عام ٢٠٠٦، و هذا ما أدّى بالنتيجة إلى ظهور الحاجة لتطوير الأدوات و اللغات البرمجية التي كانت بين يدي مطوري و مصممي مواقع الـ Web و التي أصبحت قاصرةً عن تحقيق متطلبات العملاء الطموحين بامتلاك مواقع ويب عصرية و التي سرعان ما قام من يهمه أمر التقنية بتطويرها و تقديم إصدارات جديدة منها أقوى و أكثر تأقلماً مع Web اليوم، ذلك لأنّ "التأقلم يعتبر أساس النجاح في عالم الـ Web".

يناقش هذا الكتاب أحدث إصدار لكل من لغتي HTML و CSS اللتان تعتبران الخطوة الأولى لأي مُطَوِّر Web إذ يقوم بشرح لغة HTML5 بعد التمهيد لذلك من خلال شرح HTML التقليدية من الصفر و من ثمَّ معايير XHTML من الصفر أيضاً، كما يقدِّم هذا الكتاب شرحاً مفصَّلاً لـ CSS3 بعد التمهيد لذلك من خلال شرح CSS التقليدية من الصفر أيضاً، و بهذا فإن هذا الكتاب يتألف من فصلين:

١- HTML5: يناقش هذا الفصل الإصدار الخامس (و الأحدث) من لغة البرمجة النصيَّة التشعبيَّة HTML بشكل مفصَّل و متدرَّج و يضمن لقارئه أن يكون قادراً على كتابة صفحات ويب بالمحتوى الذي يرغبه و بالشكل الذي يريده بإذن الله.

٢- CSS3: يناقش هذا الفصل كامل المزايا التي يقدمها الإصدار الثالث (و الأحدث) من أوراق الأنماط الانسيابيَّة CSS بشكل بسيط و مفهوم، و يعد قارئه أن يكون قادراً على تجميل المحتوى الذي قدَّمه في صفحات الويب التي تعلَّم كتابتها في الفصل الأوَّل.

٣- كي يكتمل الحديث عن البرمجة للويب من طرف العميل Client Side كان يجب أن يكون هناك فصل ثالث يتحدث عن لغة Java Script و لكن و لأنني من أنصار الحداثة (و السهولة) فيما يتعلق بالتقنيات البرمجية فأنصح القارئ العزيز بقراءة كتابي "تعلَّم jQuery في ١٢٠ دقيقة" أو أي مرجع يتحدث عن لغة jQuery بشكل واضح و مبسَّط بعد قراءة هذا الكتاب أولاً، لأن jQuery هي البديل العصري للغة Java Script برأيي.

أمّا عن طريقة عرض الأفكار في هذا الكتاب فقد حاولت و اجتهدت كي أجعلها غير مملَّة و لا معقَّدة و لا شبيهة بالمحتوى المتعب الذي كثيراً ما عانيت منه في الكتب التي تناقش لغات البرمجة للأسف و يمكنني أن أقول أن هذا الكتاب لا يفجِّر عشرات الأسطر البرمجيَّة غير المفهومة دفعةً واحدةً في وجه قارئه دون توضيح لها، كما أنَّه لا يطيل أكثر من المطلوب في عرض الفكرة، لكنه باختصار يقول ما يجب قوله فقط دون أن يغفل أي جانب من الجوانب التي تتوجب معرفتها.

ختاماً أسأل الله أن يتقبَّل هذا العمل و أن يبارك فيه و يكتبه صدقةً جاريةً من باب العلم الذي ينتفع به و أن يلهم قارئه الفهم و الحفظ و الصبر على التعلُّم.

م.مختار سيِّد صالح

البوكمال ٢٠١١/٨/٢٨

الفصل الأوَّل

HTML 5

مقدمة

يناقش هذا الفصل لغة HTML5 التي تعتبر حجر الأساس في إنشاء مواقع الويب، و لأنّنا لا يمكن أن نتحدث عن HTML5 مباشرةً فسيناقش هذا الفصل لغتي HTML و XHTML أولاً، و البداية مع لغة HTML.

ما هي HTML ؟

إنَّ HTML هي اختصار abbreviation الحروف الأولى من الجملة Hyper Text Markup Language و التي تعني بترجمة غير حرفية (لغة وصف صفحات الويب)[١]، و هي ليست لغة برمجيَّة، لا، بل هي لغة وصفيَّة Markup Language، لأنَّ HTML تستخدم مجموعة من الوسوم Tags لوصف صفحة الويب.

أمَّا الوسوم Tags فهي كلمات أو أحرف محددة مسبقاً Keywords تحمل معاني خاصَّة، و تكون محصورة بين قوسين من الشكل < > مثل و <html>، تأتي الوسوم في الغالب على شكل أزواج مثل و ، يدعى الوسم الأوَّل بوسم البداية Begin Tag أو وسم الفتح Opening Tag، و يدعى الوسم الثاني بوسم النهاية End Tag أو وسم الإغلاق Closing Tag.

يتم كتابة مستندات HTML باستخدام أي محرر نصي بسيط (مثل notepad) أو متقدِّم (مثل DreamWaver أو Visual Studio)، و يتم تخزينها في ملفات تحمل امتداداً extension من الامتدادين التاليين : html. أو htm.، و بالطبع لا يوجد أي فرق بينهما.

[١] لا أعرف إن قام أحد قبلي بترجمتها بهذه الطريقة لكنّني أشعر أنها الترجمة الأنسب.

تتكون مستندات HTML بشكل أساسي من محتوى نصي عادي بالإضافة إلى مجموعة من وسوم HTML، و في الحقيقة فإن مستندات HTML تدعى بـ(صفحات الويب) Web Pages، و يتم استخدام برنامج خاص لقراءة هذه المستندات يعرف هذا البرنامج باسم مستعرض الويب Web Browser و من أمثلته Internet Explorer و Firefox و Google chrome .. إلخ.

الهدف من مستعرض الويب هو قراءة مستندات HTML و عرضها بشكل صفحات ويب، بمعنى أن مستعرض الويب لا يعرض وسوم HTML كنصوص بشكل مباشر و لكنَّه يستخدمها لإنتاج صفحة ويب بالشكل و المحتوى الموصوفين في مستند HTML.

كيف أبدأ في تعلم كتابة مستندات HTML ؟

كل ما تحتاجه لكي تبدأ في تعلم كتابة مستندات HTML هو محرر نصوص (مثل Notepad) و مستعرض ويب (مثل Internet Explorer)، بعد ذلك يمكنك قراءة الفقرات التالية و التعلم خطوة بخطوة و أؤكد لك أنَّ أفضل طريقة للتعلم –برأيي– هي كتابة الأمثلة التي ستصادفك لاحقاً بشكل مباشر باستخدام محرر النصوص و من ثمَّ اختبارها باستخدام مستعرض الويب.

المثال الأوَّل في لغة HTML

سنقوم الآن بكتابة أول صفحة HTML معاً باتباع الخطوات التالية:

نقوم أولاً بفتح أحد محررات النصوص و ليكن المفكرة Notepad على سبيل المثال و نكتب الشيفرة التالية:

```
<html>

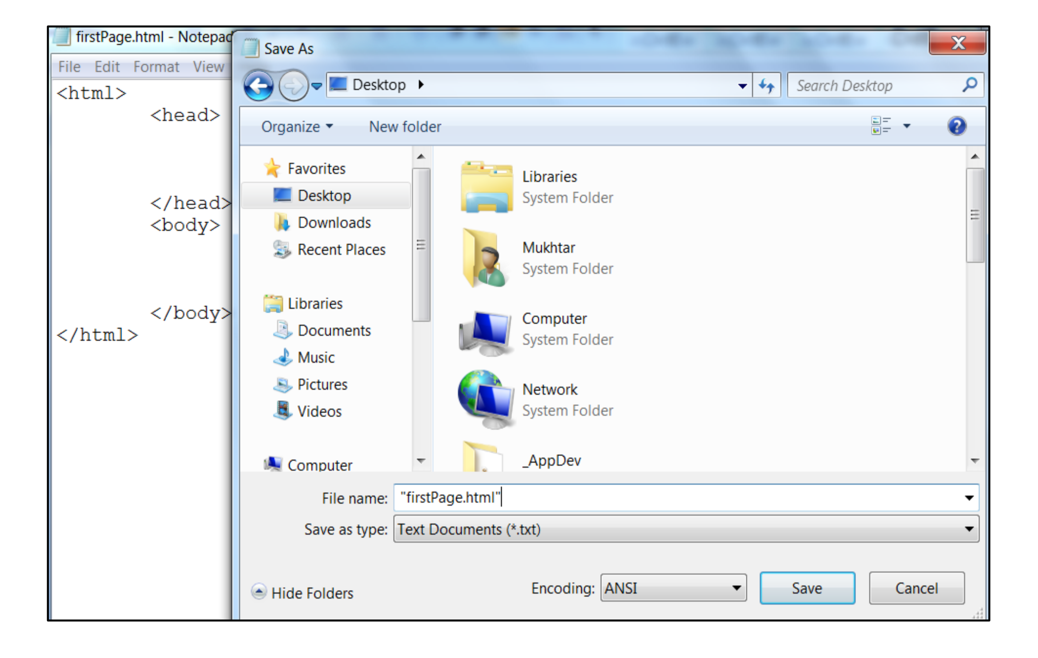
      <head>

            <title>

                  My first HTML page
```

```
            </title>
        </head>
        <body>
                <p>
                    Welcome to my first HTML page!
                 </p>
        </body>
</html>
```

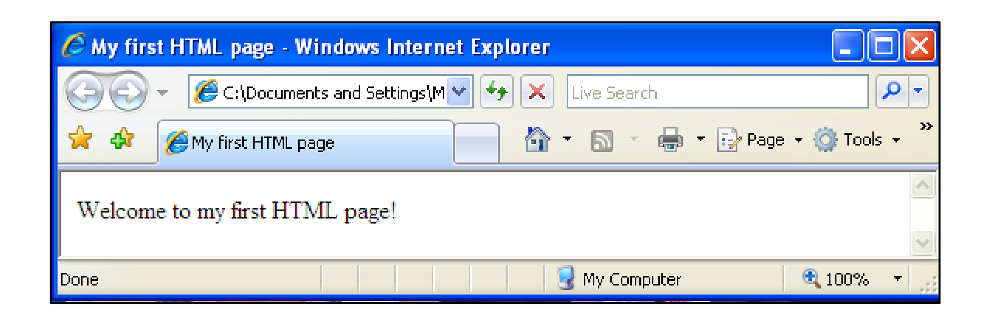
ثمَّ نقوم بحفظ الملف باسم firstPage.html



الشكل ١ : كتابة أول شيفرة HTML في برنامج المفكرة و حفظ الملف

ملحوظة : لحفظ الملف بلاحقة مختلفة عن اللاحقة .txt هناك طريقتان، الأولى أن نختار الخيار جميع الملفات *.* من القائمة المنسدلة حفظ كـ Save as type الظاهرة في الصورة أسفل اسم الملف ثم نكتب اسم الملف و لاحقته بشكل عادي، أمّا الثانية و التي أفضّلها شخصيّاً فهي أن نضع اسم الملف مع لاحقته المطلوبة بين علامتي اقتباس مزدوجتين و نختار حفظ كما هو موضّح في الصورة أعلاه.

الآن و بعد أن قمنا بحفظ الملف بالاسم firstPage.html نقوم باستعراضه باستخدام أحد مستعرضات الويب و ليكن Internet Explorer 8 على سبيل المثال لنشاهد الشكل النهائي للصفحة كمايلي:



الشكل ٢ : الشكل النهائي في المستعرض لأوَّل صفحة مكتوبة باستخدام HTML

كما ترى فقد قمنا بكتابة صفحتنا الأولى باستخدام لغة HTML و هذه الصفحة تحتوي على نص ترحيب بسيط هو النص !Welcome to my first HTML page، تم توليد هذا النص باستخدام الوسم الخاص p و بتصوّري فإنَّ هذه الصفحة على بساطتها ستكون الباب الواسع للدخول إلى لغة HTML إنْ قرأت الفقرات التالية بقليل من التركيز.

عناصر HTML

تتكون كل صفحة من صفحات HTML من مجموعة من العناصر Elements يتم إنشاؤها باستخدام الوسوم Tags و الوسوم كما قلنا سابقاً عبارة عن أحرف أو كلمات تحمل معاني خاصة بالنسبة لمستعرض الويب و غالباً ما تكون أسماء الوسوم عبارة عن اختصارات لكلمات إنجليزيَّة فالوسم <p> مثلاً يستخدم لإنشاء العنصر المسؤول عن عرض مقاطع النصوص Paragraphs ضمن صفحات HTML و الذي سيتم مناقشة عمله مع عمل باقي الوسوم بشكل تفصيلي في الصفحات التالية.

تحصر الوسوم بين أقواس من الشكل < > و من أمثلتها الوسم <html> و الوسم <head> و الوسم <title> و الوسم <body> و الوسم <p>.

لكل عنصر من عناصر HTML وسم بداية Begin Tag و وسم نهاية End Tag تكون وسوم البداية محصورة بين القوسين < > أمَّا وسوم النهاية فتكون محصورة بين القوسين السابقين مضافاً إليهما رمز الخط المائل / المعروف بـ Slash بالشكل < /> و من أمثلة وسوم النهاية الوسم </html> و الوسم </head> و الوسم </title> و الوسم </body> و الوسم </p> و يحوي كل عنصر من عناصر HTML بين وسمي بدايته و نهايته نصَّاً عاديَّاً لعرضه (كما فعل العنصر p في مثالنا السابق) و يمكن أن يحوي أي عدد آخر من عناصر HTML شرط الالتزام بترتيب متناظر لوسوم النهاية و البداية، فعند كتابة وسوم النهاية يجب أن يُراعى الترتيب فيتم كتابة وسم النهاية الخاص بوسم البداية غير المنتهي (الذي ليس له وسم نهاية) الأقرب فالأقرب، ففي حين أن الترتيب التالي يعتبر صحيحياً:

```
<html>
      <body>
            <p>
              نص عادي سيتم عرضه في المستعرض!
             </p>
      </body>
</html>
```

لأنَّ وسم النهاية الأوَّل </p> جاء لإنهاء أقرب وسم بداية غير منتهي و هو <p> في مثالنا، ثمَّ جاء وسم النهاية </body> لإنهاء أقرب وسم بداية غير منتهي و هو <body>، ثمَّ جاء وسم النهاية </html> لإنهاء أقرب وسم بداية غير منتهي و هو <html>.

فإنَّ الترتيب التالي يعتبر خاطئاً:

```
<html>
      <body>
```

```
            <p>
                  نص عادي سيتم عرضه في المستعرض!
            </body>
      </html>
</p>
```

لأن وسم النهاية </body> جاء لإنهاء الوسم <body> الذي لم يكن الوسم الأقرب غير المنتهي للأسف و الحالة نفسها تكررت مع وسمي النهاية </html> و </p>.

الخلاصة: لا تنسَ كتابة وسم الإغلاق و راعِ ترتيب وسوم الإغلاق.

الفراغات White Spaces

في لغة HTML لا يوجد أي قيمة لمحارف الفراغات (مثل Space و Tab .. إلخ) عند استعراض الصفحات باستخدام مستعرض الويب و إنَّما تستخدم محارف الفراغات لغرض ترتيب الشيفرة و جعلها مقروءة بشكل أوضح فقط، فقراءة شيفرة HTML التالية على سبيل المثال:

```
<html>
      <body>
            <h1>
                  Hi!
            </h1>
      </body>
</html>
```

أسهل بكثير من قراءة شيفرة HTML التالية:

```
<html><body><h1>Hi!</h1></body></html>
```

مع أنَّهما متماثلتان تماماً عند استعراضهما باستخدام مستعرض الويب.

فائدة : من العادات البرمجيَّة الجيدة إزاحة كل المحتوى الموجود بين وسمي بداية و نهاية متماثلين بمقدار ضغطة Tab و البعض يفضِّل إزاحتها بمقدار ثلاث فراغات (ثلاث ضغطات على مفتاح Space).

الشكل العام لعناصر HTML

يتم وصف عناصر HTML باستخدام الوسوم، و فيمايلي نعرض الشكل العام لعنصر HTML:

- يبدأ عنصر HTML بوسم البداية Begin Tag.
- ينتهي عنصر HTML بوسم النهاية End Tag.
- كل ما يتم كتابته بين وسمي البداية و النهاية يدعى محتوى عنصر HTML علماً أن هناك بعض العناصر التي لا تحوي أي محتوى و التي تسمى عديمة المحتوى.
- يندمج وسما البداية و النهاية في وسم واحد في حالة العناصر عديمة المحتوى.
- يتم تمرير مجموعة من الخصائص لأغلب عناصر HTML في وسم البداية عن طريق الواصفات Attributes.

ملحوظة: دائماً و أبداً، قم بكتابة عناصر HTML بالأحرف الإنجليزيَّة بحالتها الصغيرة Lower Case.

لنشاهد الأمثلة التالية:

وسم النهاية	محتوى العنصر	وسم البداية
</p>	Welcome to my website.	<p>
****	Go to index	<a href="index.htm">
		<hr />

يمثِّل كل سطر في الجدول السابق عنصراً من عناصر HTML في الصفحة بشكل تخيُّلي، فالعنصر الأوَّل يبدأ بالوسم <p> و ينتهي بالوسم </p> و يحوي المحتوى النصي Welcome to

my website بينما يبدأ العنصر الثاني بالوسم و ينتهي بالوسم و يحوي المحتوى النصي Go to index مع إسناد القيمة index.htm إلى واصفته href، في حين أنَّ العنصر الثالث عديم المحتوى يتكون من اتحاد وسمي البداية و النهاية في وسم واحد بالشكل <hr />.

الشكل العام لصفحات HTML

تتكون كل صفحة HTML من ثلاث مناطق:

- منطقة جسد الصفحة Body Section:

و هي المنطقة المحصورة بين وسمي <body> و </body> و هذه المنطقة هي التي تنتج الشكل النهائي للصفحة و تضم جميع العناصر التي تمثِّل المحتوى الظاهر للصفحة و الذي سيظهر في مستعرض الويب عند استعراض هذه الصفحة و بالطبع فإن الجزء الأكبر من عناصر HTML سيكون في هذه المنطقة.

- منطقة رأس الصفحة Head Section:

و هي المنطقة المحصورة بين وسمي <head> و </head> و هذه المنطقة تحوي مجموعة من عناصر HTML أغلبها لا يظهر في مستعرض الويب عند استعراض الصفحة و لكن المهمة الرئيسية لهذه المنطقة هي إعطاء معلومات عن ماهيَّة المحتوى الموجود بالصفحة إضافة لبعض المعلومات غير الظاهرة الأخرى و التي تستخدمها محركات البحث من أجل الأرشفة و البحث في الغالب.

- المنطقة الأم[٢]:

[٢] لم يرد هذا المصطلح في أي من المراجع التي قرأتها و إنَّما هو اجتهاد شخصي لتبسيط الفكرة المطروحة (برأيي).

و هي المنطقة المحصورة بين وسمي <html> و </html> و هذه المنطقة هي المنطقة التي تحدد بداية و نهاية الصفحة و هي التي تضم منطقتي الرأس و الجسد و بهذا فهي تضم كامل مستند (صفحة) HTML.

و بهذا يكون الشكل العام لأي مستند HTML كمايلي:

```
<html>
      <head>
          محتوى منطقة الرأس غير الظاهر سيكون هنا
      </head>
      <body>
          محتوى الصفحة النهائي و الظاهر فعليّاً سيكون هنا
       </body>
</html>
```

واصفات الوسوم Attributes

يتلخّص كل ما قلناه سابقاً بأنَّ لغة HTML تتيح لك تقديم المحتوى عبر إنشاء مجموعة من العناصر، مهمة كل عنصر من عناصر HTML عرض شيء محدد جداً في صفحتك فهناك عناصر لعرض النصوص و هناك عناصر لعرض الصور و هناك عناصر لعرض الروابط التشعبيَّة Links .. إلخ، يتم إنشاء كل عنصر من هذه العناصر بواسطة وسم خاص من وسوم HTML، و يكون لكل عنصر وسم بداية و وسم نهاية يحصران محتوى العنصر الظاهر بينهما.

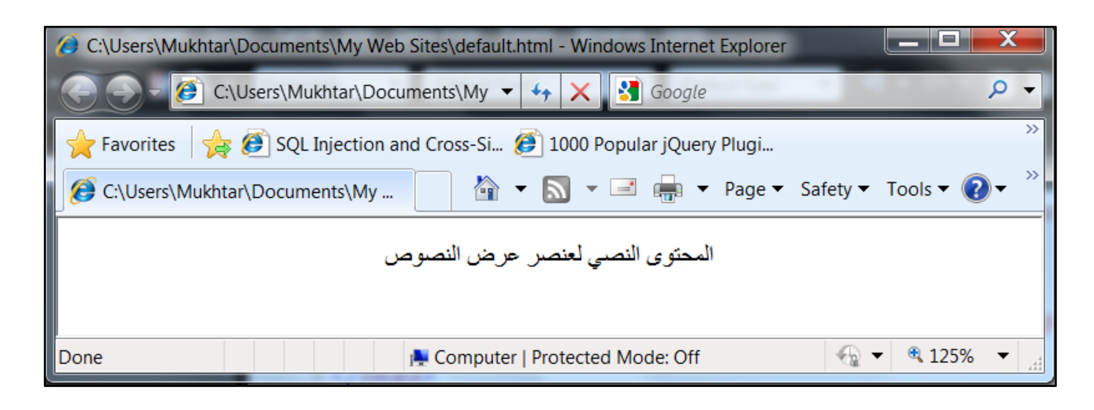
يمكن تزويد بعض وسوم البداية بمجموعة من الخصائص الإضافيَّة و التي تخصّص سلوك عرض العنصر لمحتواه و يتم هذا عبر ما يعرف بالواصفات Attributes، انظر للشيفرة التالية على سبيل المثال:

```
<p align="center">

    المحتوى النصي لعنصر عرض النصوص

</p>
```

كما تلاحظ فقد قمنا بإضافة شيء جديد في هذه الشيفرة إلى وسم البداية الخاص بالعنصر p ألا و هو الواصفة align التي تحدد محاذاة النص الذي سيعرضه العنصر p و أعطيناها القيمة center لعرض النص في منتصف الصفحة و عند استعراضها سيظهر النص كمايلي:



الشكل ٣ : عنصر عرض النصوص <p> عند إضافة واصفة المحاذاة align

حسناً، يوجد لكل وسم بداية من وسوم HTML مجموعة معروفة سلفاً من الواصفات Attributes و يوجد لكل واصفة مجموعة معروفة سلفاً من القيم التي يمكن أن يتم إسنادها إلى الواصفة، فالواصفة align هي إحدى الواصفات المعروفة سلفاً لوسم البداية <p> على سبيل المثال و القيم التي يمكن أن تسند إليها هي left لمحاذاة النص إلى اليسار أو right لمحاذاة النص إلى اليمين أو center لمحاذاة النص إلى الوسط، و سيتم بالطبع عرض واصفات كل وسم بالتفصيل عند الحديث عنه و أظن أنه لا داعي للقول أن الشكل العام للواصفة هو name="value" حيث أنَّ name هو اسم الواصفة و value هي القيمة المُسنَدَة لتلك الواصفة و التي يجب أن توضع بين علامتي اقتباس مزدوجتين " " أو علامتي اقتباس مفردتين ' '.

ملحوظة : يجب أن تتم كتابة وسوم و واصفات HTML بحروف إنجليزيَّة صغيرة Lower Case.

أظن أنَّنا نستطيع الآن أن نبدأ بالحديث عن عناصر HTML كافةً و بالتفصيل و البداية مع عناصر العناوين Headings، و لكن قبل ذلك سنعرض جدولاً بمجموعة من الواصفات المشتركة بين أغلب وسوم HTML هنا بدلاً من تكرارها مع كل وسم من تلك الوسوم، و فيمايلي الجدول:

اسم الواصفة	القيم الممكنة	الشرح
id	اي اسم فريد(غير مكرر) بالنسبة للمستند	يتم استخدام هذا الاسم للتعامل مع العنصر برمجيَّاً باستخدام لغة Java Script أو jQuery، و لها استخدامات أخرى سيتم عرض أحدها في فقرة الروابط الداخلية لاحقاً
dir	ltr rtl	لتحديد اتجاه القراءة، ltr تعني أن اتجاه القراءة من اليسار إلى اليمين أمَّا rtl فتعني أن اتجاه القراءة من اليمين إلى اليسار
align	left right center justify	لتحديد محاذاة النص
class	أي اسم فئة CSS صالح	لمنح كافة خصائص الفئة إلى العنصر و سيتم مناقشة هذا بالتفصيل في الفصل الخاص بـ CSS3
name	أي اسم فريد	تستخدم لتمييز العنصر برمجيّاً

الجدول ١ : جدول الواصفات المشتركة بين أغلب عناصر HTML

عناصر العناوين Headings

توفر لغة HTML ستَّة عناصر لعرض العناوين و هي على الترتيب: h1 و h2 و h3 و h4 و h5 و h6 حيث أنَّ العنصر h1 هو أكبرها حجماً و العنصر h6 هو الأصغر و ما بينهما يتدرج في الحجم، و طبعاً حرف الـ h هنا اختصار لكلمة Heading، دعنا نجرب الشيفرة التالية على سبيل المثال:

```
<html>
	<head>
		<title>
			صفحة اختبار لعناصر العناوين
		</title>
	</head>
	<body dir='rtl'>
		<h1>عنصر عنوان من المستوى الأوَّل</h1>
		<h2>عنصر عنوان من المستوى الثاني</h2>
		<h3>عنصر عنوان من المستوى الثالث</h3>
		<h4>عنصر عنوان من المستوى الرابع</h4>
		<h5>عنصر عنوان من المستوى الخامس</h5>
		<h6>عنصر عنوان من المستوى السادس</h6>
	</body>
</html>
```

عند استعراض الصفحة السابقة في مستعرض الويب سنشاهد النتيجة التالية:



الشكل ٤ : صفحة اختبار عناصر العناوين h1 إلى h6

كما تلاحظ فقد قمنا بتزويد الوسم <body> بالواصفة dir و التي تحدد اتجاه قراءة جسد المستند و قمنا بإسناد القيمة rtl لها و ذلك لنجعل اتجاه القراءة من اليمين إلى اليسار right to left و هو ما يتناسب مع اللغة العربية بالطبع، و في حال أن هذه الواصفة لم تُعْطَ أية قيمة فإن القيمة الافتراضية لها هي ltr أي أن اتجاه القراءة الافتراضي من اليسار إلى اليمين.

ملحوظة: استخدم عناصر العناوين <h1> إلى <h6> من أجل عناوين الفقرات فقط، و لا تستخدمها لجعل النص عريضاً أو كبير الحجم فهناك وسوم خاصة لهذين الغرضين.

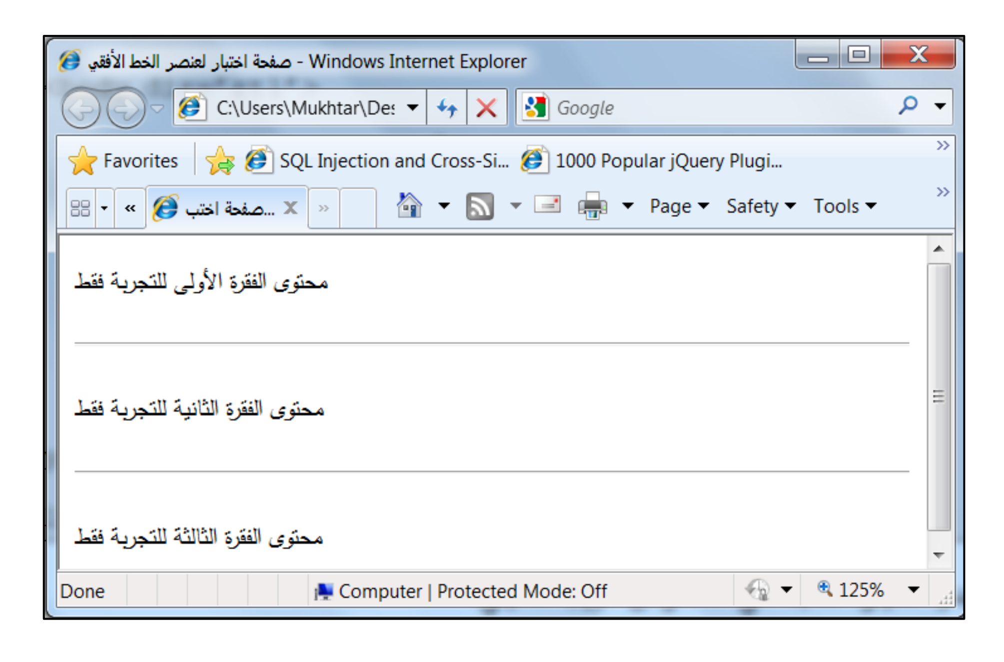
عنصر الخط الأفقي Horizontal Line

تقدِّم لغة HTML الوسم الخاص <hr /> لإنشاء الخطوط الأفقيَّة في الصفحة، انظر للمثال التالي:

```
<html>
    <head>
```

```
        <title>
            صفحة اختبار لعنصر الخط الأفقي
        </title>
    </head>
    <body dir='rtl'>
        <p>محتوى الفقرة الأولى للتجربة فقط</p>
        <hr />
        <p>محتوى الفقرة الثانية للتجربة فقط</p>
        <hr />
        <p>محتوى الفقرة الثالثة للتجربة فقط</p>
</body>
</html>
```

تبدو الشيفرة السابقة في مستعرض الويب كمايلي:



الشكل ٥ : صفحة تجربة العنصر <hr />

التعليقات Comments

اعتاد المبرمجون في لغات البرمجة التقليديَّة على كتابة أسطر توضيحيَّة لا تُعالَج إذ أنها لا تعتبر جزءاً من الشيفرة، و إنَّما تستخدم فقط لتذكير المبرمج بأجزاء الشيفرة عندما يعود لتعديلها بعد فترة من الزمن، و هذه الأسطر تعرف بالتعليقات Comments، و مع أنَّ HTML لغة وصفية كما قلت سابقاً إلّا أنها توفر آلية لكتابة التعليقات ضمن المستندات، فتقدِّم الصيغة العامة التالية لكتابة التعليق:

```
<!-- comment -->
```

لنشاهد المثال التالي على استخدام التعليقات ضمن مستند HTML:

```
<html>

     <head>

          <title>صفحة اختبار لعنصر الخط الأفقي مضافاً إليها مجموعة من التعليقات البرمجيَّة</title>

     </head>

     <body dir='rtl'>

          <!-- السطرين التاليين للفقرة الأولى -->

          <p>محتوى الفقرة الأولى للتجربة فقط</p> <hr />

          <!-- السطرين التاليين للفقرة الثانية -->

          <p>محتوى الفقرة الثانية للتجربة فقط</p> <hr />

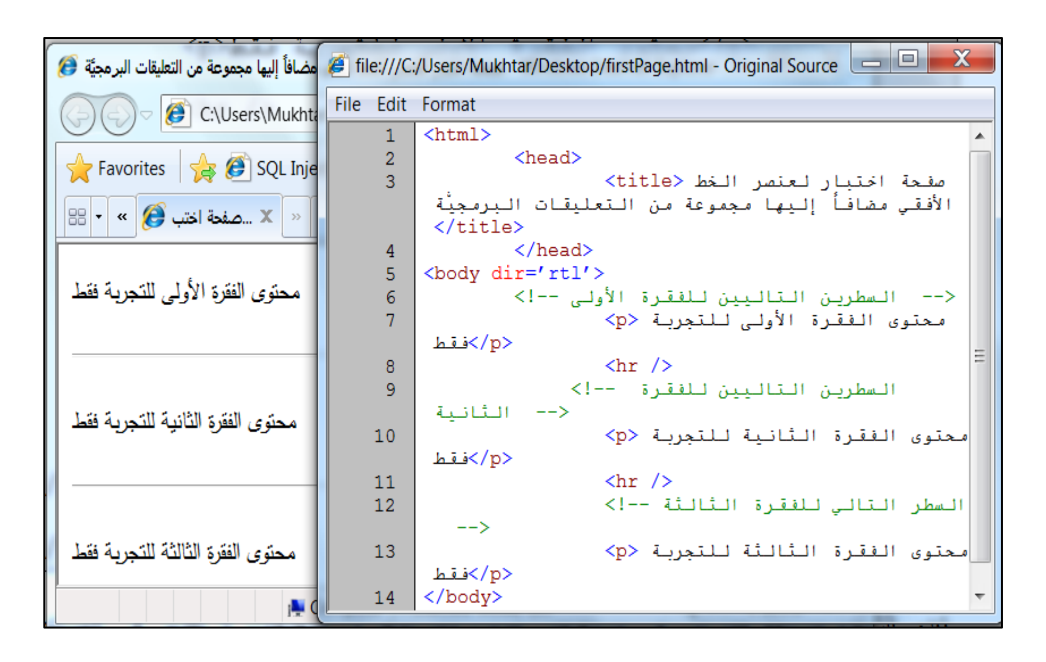
          <!-- السطر التالي للفقرة الثالثة -->

          <p>محتوى الفقرة الثالثة للتجربة فقط</p>

     </body>

</html>
```

تبدو الشيفرة السابقة كمايلي في مستعرض الويب مما يؤكّد أن التعليقات لا تظهر في المستند(أثناء استعراض الصفحة جرِّب اختيار الأمر "عرض-> المصدر" أو "View->source"):



الشكل ٦ : التعليقات تظهر باللون الأخضر في نافذة عرض المصدر و لا تظهر في الصفحة

ملحوظة: لا تنسَ إشارة التعجُّب بعد القوس الأوّل في بداية وسم التعليق.

عناصر النصوص Paragraphs

تتكون صفحة الويب في الواقع من مجموعة من الفقرات النصية في الغالب، يتم إنشاء هذه الفقرات باستخدام الوسم <p> و الذي مر معنا استخدامه في الأمثلة السابقة، و يستخدم الوسم
 للنزول إلى السطر التالي و ذلك لأن محارف الفراغات بما فيها محرف المفتاح Enter ليس لها قيمة في لغة HTML كما ذكرنا سابقاً.

عناصر تنسيق النصوص Text Formatting

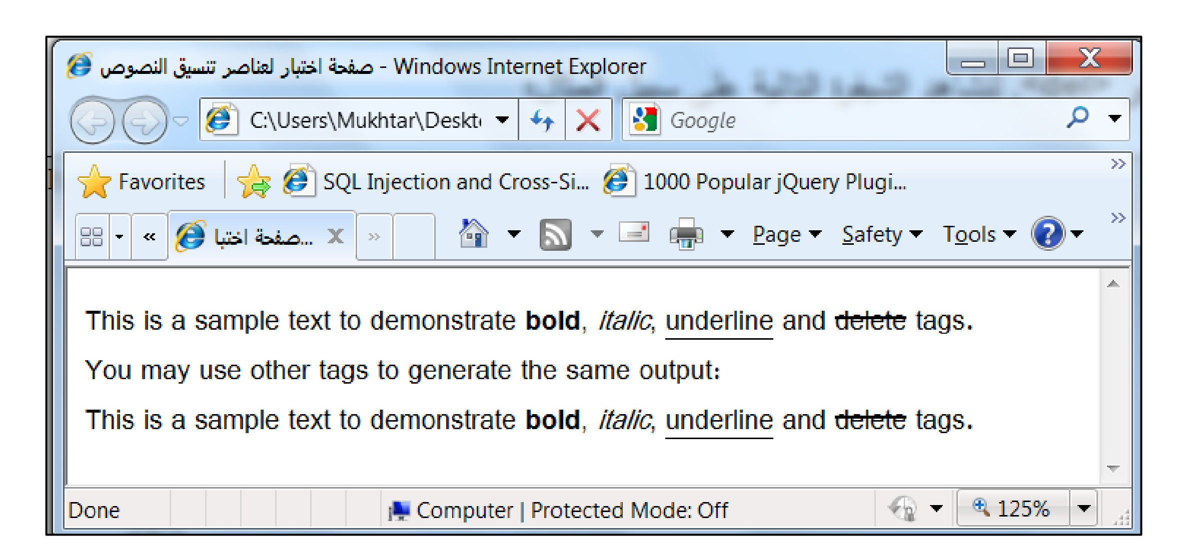
توفر لغة HTML مجموعة من العناصر لتنسيق النصوص، فلجعل النص عريضاً Bold توفر العنصر <b> (أو العنصر <strong>)، و لجعل النص مائلاً Italic توفر العنصر <i> (أو

العنصر <em>)، و لوضع خط أسفل النص توفر العنصر <u>، و لشطب النص Delete توفر العنصر <del>، لنشاهد الشيفرة التالية على سبيل المثال:

```
<html>
  <head>
     <title>صفحة اختبار لعناصر تنسيق النصوص</title>
  </head>
  <body>
     <p>
           This is a sample text to demonstrate
          <b>bold</b>,
          <i>italic</i>,
          <u>underline</u> and
          <del>delete</del> tags.
           <!-- استخدمنا <br/> للنزول سطراً واحداً -->
           <br/>
           You may use other tags to generate the same output:
          <br/>
          This is a sample text to demonstrate
          <strong>bold</strong>,
          <em>italic</em>,
          <u>underline</u> and
          <del>delete</del> tags.
      </p>
  </body>
```

```
</html>
```

تبدو الشيفرة أعلاه في المستعرض كمايلي:



الشكل ٧ : صفحة اختبار عناصر تنسيق النصوص

ملحوظة: بالطبع هناك المزيد من عناصر تنسيق النصوص سيتم ذكرها في مرجع وسوم HTML في نهاية هذا الفصل.

المحارف و الرموز الخاصَّة Special Characters

هناك بعض المحارف و الرموز التي لا يمكن عرضها في الصفحة عن طريق كتابتها بشكل مباشر مثل الرموز المستخدمة في الرياضيات و محارف الفراغات و غيرها، توفر لغة HTML آلية خاصة لعرض هذه الرموز و ذلك باتباع الصيغة العامة التالية &value; باستبدال القيمة value بقيمة المحرف المطلوب أن يتم عرضه في مستعرض الويب و في الجدول التالي نماذج من هذه المحارف:[٣]

[٣] للحصول على الجدول الكامل يمكن اتباع الرابط التالي:

http://www.w3schools.com/tags/ref_symbols.asp

الرمز	شيفرة XHTML
<	<
>	>
≠	≠
™	™
©	©
محرف الفراغ	

الجدول ٢ : بعض المحارف الخاصة في HTML

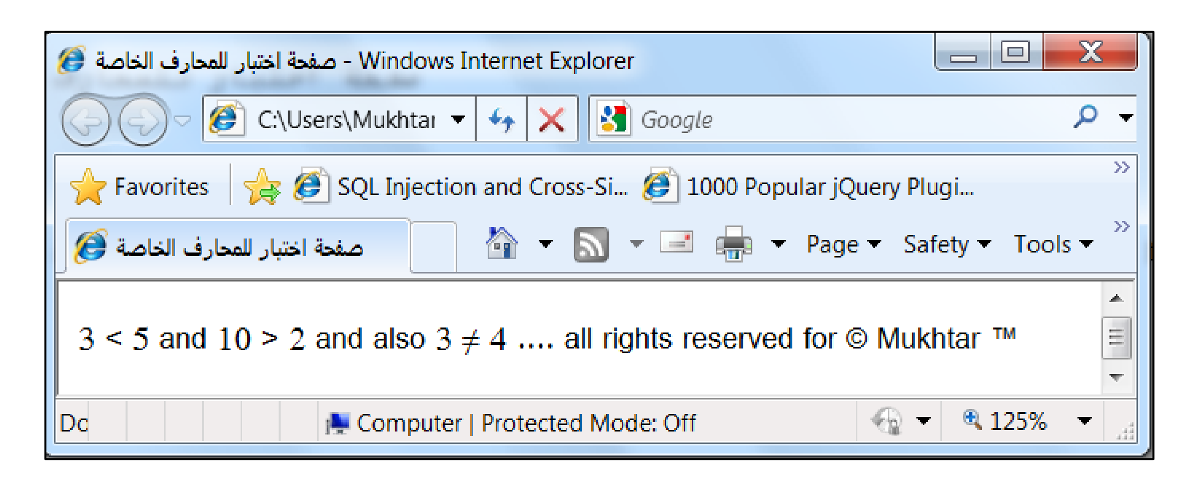
لنشاهد المثال التالي:

```
<html>
  <head>
     <title>
           صفحة اختبار لبعض المحارف الخاصة
     </title>
  </head>
  <body>

     <p>
       3 &lt; 5 and 10 &gt; 2 and also 3 &ne; 4 ....
       all rights reserved for &copy; Mukhtar &trade;
     </p>

  </body>
</html>
```

و الذي يبدو عند عرضه في المستعرض كمايلي:



الشكل ٨ : مثال لاستخدام بعض المحارف الخاصة في الصفحة

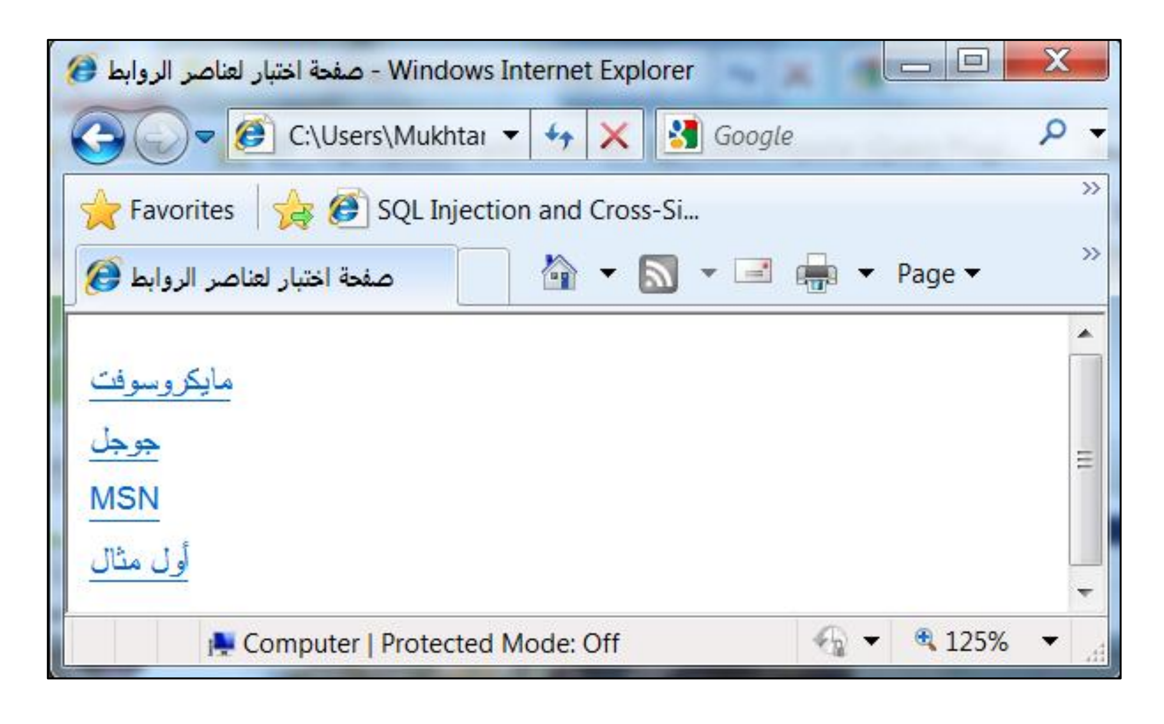
عناصر الروابط Hyper Links

توفر لغة HTML آلية للانتقال بين الصفحات المختلفة عبر عناصر الروابط Hyper Links، و الروابط عبارة عن نصوص أو صور تنقلك عند النقر عليها من الصفحة الحالية إلى إلى صفحة ويب أخرى، تقوم مستعرضات الويب بتمييز الروابط بعرض "يد صغيرة" كمؤشر للفأرة عند الإشارة إلى أحد الروابط، و يتم إنشاء الروابط بواسطة الوسم <a>، و الـ a هذه اختصار لـ Anchor ، يتم تزويد عنصر الرابط بنص يُعرض كمحتوى له أمّا الموقع الهدف (الذي سيتم الانتقال إليه عند النقر على الرابط) فيتم تزويده للواصفة href و التي هي اختصار لـ Hyper Reference، فللنظر للمثال التالي:

```
<html>
  <head>
     <title>
          صفحة اختبار لعناصر الروابط
     </title>
  </head>
```

```
  <body>
    <a href="http://www.microsoft.com">مايكروسوفت</a><br />
    <a href="http://www.google.com">جوجل</a><br />
    <a href="http://www.msn.com" target="_blank" >MSN</a>
    <br />
    <a href="firstPage.html" target="_self">أول مثال</a>
  </body>
</html>
```

الذي يبدو عند استعراضه بمستعرض الويب كمايلي:



الشكل ٩ : صفحة اختبار عنصر الروابط

لقد قمنا بتزويد صفحتنا بمجموعة من عناصر الروابط التي تشير إلى بعض المواقع الشهيرة و قد تعمدت في الرابط الأخير ذكر اسم ملف المثال الأوّل الذي قمنا بإنشائه معاً في بداية هذا الفصل و ذلك لتوضيح أن الروابط يمكن أن تكون مطلقة Absolute (كالثلاثة الأولى) أو نسبية Relative

(كالرابط الأخير) و يقصد بالنسبية أن الرابط المذكور يكون عبارة عن مسار Path ملف الصفحة الهدف بالنسبة للصفحة الحالية و في حالتنا فقد كانت الصفحة الهدف firstPage.html في نفس مجلد الصفحة الحالية [٤].

عند النقر على أحد هذه الروابط سيتم الانتقال إلى الرابط URL المذكور في الواصفة href المرافقة لعنصر الرابط الذي تم النقر عليه، من الجدير بالذكر أنَّنا قمنا بتزويد بعض الروابط بالواصفة target و التي تحدد مكان عرض الصفحة الهدف في المستعرض بمعنى أنّها تجيب على الأسئلة التالية:

هل سيتم عرض الصفحة الهدف في نفس الصفحة الحالية؟ (في هذه الحالة فإن قيمة الواصفة هي القيمة _self)، هل سيتم عرض الصفحة الهدف في نافذة مستعرض جديدة؟ (في هذه الحالة فإن قيمة الواصفة _blank)، هل سيتم عرض الصفحة الهدف في إطار محدد؟ (في هذه الحالة فإن قيمة الواصفة هي اسم الإطار و سيتم تناول الإطارات بشكل مفصّل لاحقاً).

آخر ما يجب ذكره عن الروابط حالياً أنَّه يمكن للرابط أن يشير إلى أي مورد Resource متوفر على الويب و ليس فقط الصفحات، كما يمكن أن يشير الرابط إلى بريد إلكتروني E-Mail و ذلك بأن نضع القيمة :mailto قبل البريد الإلكتروني الذي نريد أن نضع رابطاً له، انظر المثال التالي:

```
<html>
  <head>
     <title>
         روابط خاصة
     </title>
```

[٤] يمكنك تعلم المزيد عن المسارات Paths من خلال زيارة الموقع:

http://msdn.microsoft.com/en-us/library/aa365247(VS.85).aspx

```
  </head>

  <body>

     <a href="http://www.somesite.com/files/file1.zip"> انقر هنا لتحميل الملف</a>

     <br />

     <a href ="mailto:mokhtar_ss@hotmail.com">راسلني</a>

  </body>

</html>
```

ما سيحدث عند النقر على الرابط الأوّل هو ظهور مربّع تحميل الملف Save File Dialog و ما سيحدث عند النقر على الملف الثاني هو ظهور برنامج مدير البريد الإلكتروني (مثل Outlook Express) في وضع إرسال رسالة جديدة إلى البريد الإلكتروني المذكور في الرابط.

عناصر الصور Images

توفر لغة HTML عنصراً خاصاً لعرض الصور ضمن الصفحة هذا العنصر هو العنصر <img/> (اختصاراً لـ Image) الذي يقوم بعرض الصورة ضمن الصفحة اعتماداً على مسارها الذي يمرر للعنصر عبر الواصفة src (اختصاراً لـ Source)، و كما هو الحال بالنسبة للعنصر الخاص
 فإنّ العنصر <img/> لا يمتلك أي وسم نهاية و لذلك فإنّ شكله العام في الغالب يكون ، لنشاهد المثال التالي:

```
<html>

  <head>

     <title>

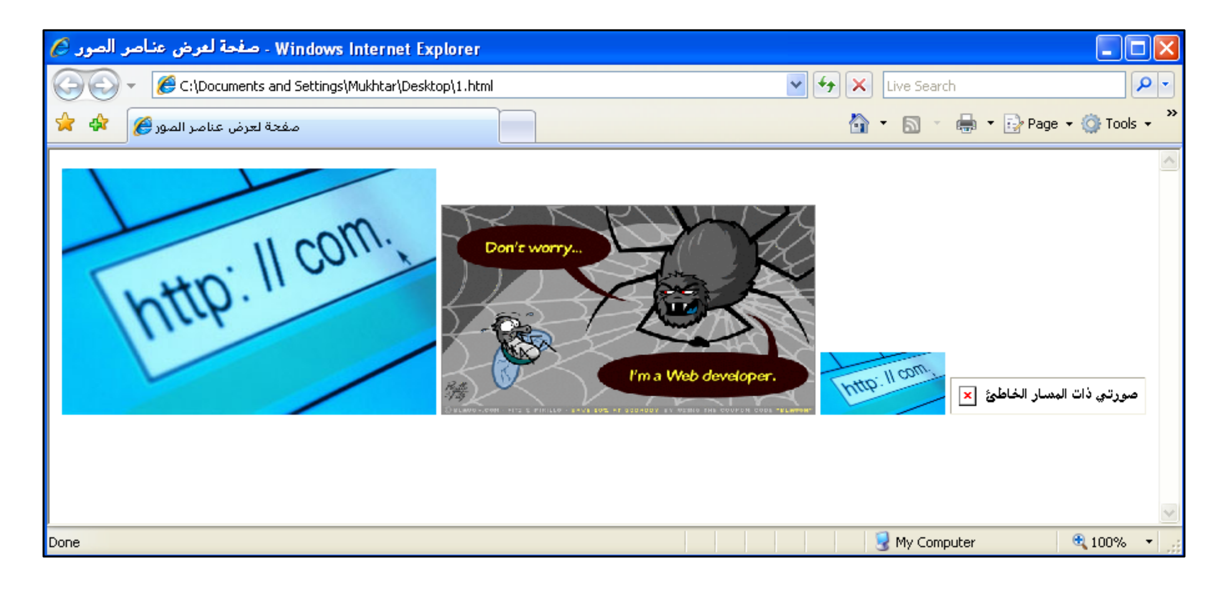
           صفحة لعرض عناصر الصور

      </title>

  </head>
```

```
  <body>
     <img src="myPic.jpg" />
      <img src="myPic2.gif" alt="الصورة الثانية" />
      <img src="myPic.jpg" alt="الصورة الثالثة" width="100px"
height="50px" />
      <img src="مسار خاطئ" alt="صورتي ذات المسار الخاطئ" />
  </body>
</html>
```

و الذي يبدو عند عرضه في المستعرض كمايلي:



الشكل ١٠ : صفحة اختبار لعنصر الصور

كما تلاحظ فقد قام عنصر عرض الصور بعرض الصور ذات المسارات المذكورة في كل وسم <img/>، و من واصفات هذا العنصر الواصفة alt التي تسند إليها قيمة نصيّة يتم عرضها في حال تعذر الوصول إلى الصورة المذكورة في الواصفة src و هذه الحالة حدثت معنا في الصورة الرابعة و التي زودناها بمسار خاطئ عن قصد لعرض هذه الحالة، كما يمكن عرض الصورة بمقاس محدد و ذلك عبر تزويد العنصر img بواصفتي الارتفاع height و العرض width و التي

تسند إليهما قيم رقميَّة تحدد عرض و ارتفاع الصورة المعروضة بالبكسل(هذا ما تعنيه px في القيم المسندة لهاتين الواصفتين).

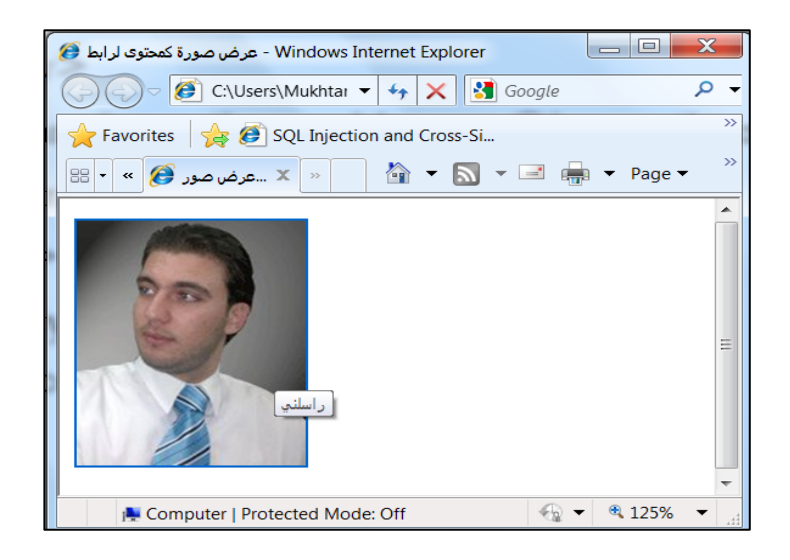
ملحوظة: على فرض أن صفحتنا تحتوي ٥ صور فإنَّ عرض الصفحة بشكل تام و صحيح سيحتاج لتحميل ٦ ملفات هي ملف الصفحة و ملفات الصور الخمسة و هذا ما قد يسبب حملاً زائداً على موقعك مستقبلاً في حال الإفراط في استخدام الصور.

الصور بدلاً من نصوص الروابط

من الشائع في مواقع الويب أن يتم استبدال نصوص محتوى عناصر الروابط <a> بالصور و ذلك لتجميل محتوى الموقع بحيث يتم الانتقال للصفحة الهدف للرابط عند النقر على الصورة بدلاً من النقر على النص و يتم ذلك بتضمين العنصر <img/> كمحتوى للعنصر <a> و فيمايلي نعرض مثالاً بسيطاً لذلك:

```
<html>
  <head>
     <title>
          عرض صورة كمحتوى لرابط
     </title>
  </head>
  <body>
     <a href = "mailto:mokhtar_ss@hotmail.com">
        <img src="myPic.jpg" alt="راسلني" />
     </a>
  </body>
</html>
```

و هذا ما يبدو في المستعرض كمايلي:



الشكل ١١ : استخدام الصور بدلاً عن النصوص كمحتوى للروابط

الخرائط الصوريَّة Image Maps

الخرائط الصوريَّة تعني جعل أجزاء محدّدة من صورة ما روابطاً، و يتم هذا على ثلاث مراحل، الأولى هي أن يتم وضع الصورة في الصفحة بشكل تقليدي عن طريق الوسم <img/>، أما الثانية فهي إنشاء عنصر الخريطة <map> مع إسناد قيمة الواصفة name الخاصة به كاسم للخريطة و من ثمَّ إنشاء مجموعة من عناصر <area> بإحداثيات المناطق التي ستتحول إلى روابط في الصورة، و المرحلة الثالثة تتم بربط عنصر الخريطة و عنصر الصورة و ذلك بإسناد اسم الخريطة إلى الواصفة usemap الخاصة بعنصر الصورة، لنشاهد المثال التالي:

```
<html>

<body>

      <p>انقر على أحد الكواكب لعرض معلومات عنه</p>

      <img src="planets.gif" width="145" height="126"
alt="الكواكب" usemap="#planetmap" />

      <map name="planetmap">
```

```
            <area shape="rect" coords="0,0,82,126" alt="الشمس"
href="sun.htm" />

            <area shape="circle" coords="90,58,3" alt="المريخ"
href="mercur.htm" />

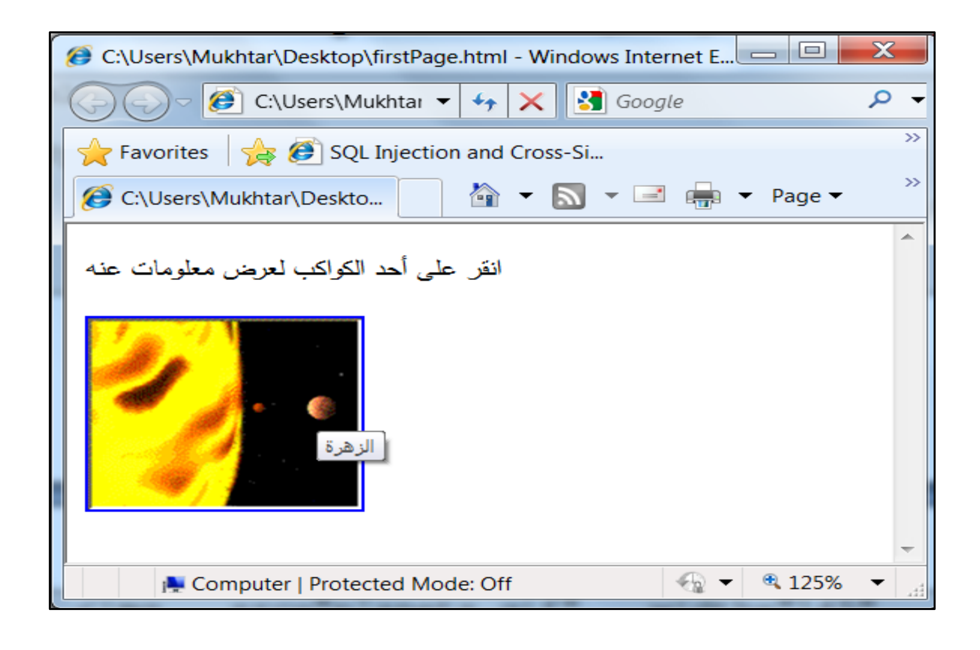
            <area shape="circle" coords="124,58,8" alt="الزهرة"
href="venus.htm" />

      </map>

</body>

</html>
```

يبدو المثال التالي في المستعرض كمايلي:

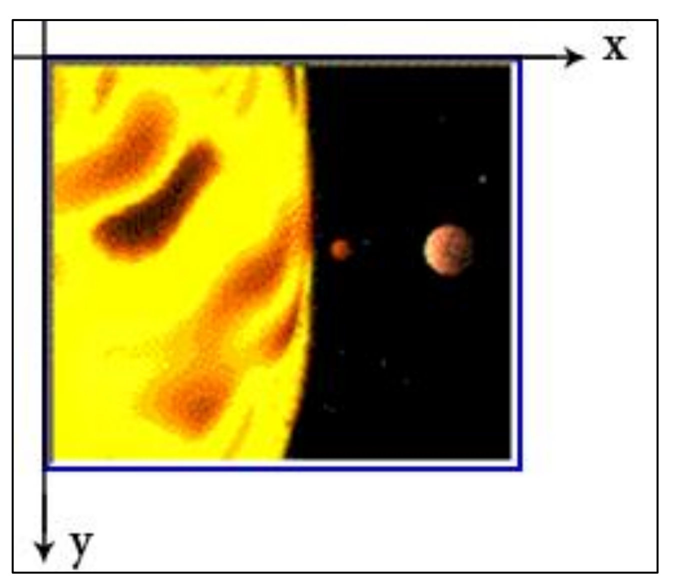


الشكل ١٢ : مثال على صنع خريطة صورية

كما تلاحظ فقد تمت إضافة الصورة بشكل طبيعي أولاً، ثمَّ بدأنا بإنشاء الخريطة و أعيطناها اسماً بالسطر:

```
      <map name="planetmap">
```

بعد ذلك قمنا بإنشاء منطقة لكل كوكب من الكواكب عبر الوسم area و الذي له واصفتان الأولى shape لتحديد شكل المنطقة (مستطيلة، دائرية .. إلخ)، أمَّا الثانية فهي واصفة الإحداثيات coords حيث تم تمرير أربعة إحداثيات في حالة المستطيل لتحديد أبعاد المستطيل يمثل الإحداثيان الأول و الثاني الركن الأيسر العلوي من المستطيل بينما يمثل الإحداثيان الثالث و الرابع الركن الأيمن السفلي من المستطيل و هذا كافٍ لتحديده، و بالطبع تم هذا على اعتبار أن مركز الجملة الديكارتيَّة للصورة هو الركن الأيسر العلوي من الصورة، و بهذا فإن النقطة (٠,٠) و النقطة (82,126) تحددان منطقة كوكب الشمس بالمستطيل المرسوم بينهما، أمَّا في حالة المنطقة الدائرية فيتم تمرير ثلاثة إحداثيات يعبر الأول و الثاني عن نقطة مركز الدائرة و يعبر الثالث عن نصف قطرها.



الشكل ١٣ : شكل تخيلي لتوضيح المبدأ الهندسي في رسم المناطق

و في النهاية قمنا بتمرير اسم الخريطة إلى الواصفة usemap الخاصة بالصورة.

ملحوظة: يتم تمرير اسم الخريطة إلى الواصفة usemap مسبوقاً بالرمز #

الروابط الداخليَّة Internal Linking

أحياناً يكون المحتوى الموجود ضمن صفحة معيّنة كبيراً إلى حدٍّ ما و في حالة مثل هذه يصبح الانتقال إلى فقرة محددة ضمن الصفحة عملية تستغرق بعض الجهد و الوقت، توفِّر لغة HTML آليَّة للانتقال إلى جزء محدد من الصفحة مباشرة عبر ما يعرف بالروابط الداخلية Internal Links

و التي يتم تحقيقها عبر استخدام وسم الرابط <a> و تمرير معرَّف العنصر المراد الانتقال إليه كقيمة للواصفة href بالشكل التالي:

```
<a href="#id"> id انتقل إلى العنصر ذو المعرف</a>
```

حيث أنّ id يمثل معرف العنصر المراد الانتقال إليه ضمن الصفحة و المثال التالي يوضِّح هذه الفكرة:

```
<html>
  <head>
      <title>
         مثال على الروابط الداخلية
      </title>
  </head>
  <body>
      <a href="#Para4">إلى الفقرة الرابعة</a>
      <h1>الفقرة الأولى</h1>
      <p>هنا محتوى الفقرة الأولى</p>
      <h1>الفقرة الثانية</h1>
      <p>هنا محتوى الفقرة الثانية</p>
      <h1>الفقرة الثالثة</h1>
      <p>هنا محتوى الفقرة الثالثة</p>
      <h1 id="Para4">الفقرة الرابعة</h1>
      <p>هنا محتوى الفقرة الرابعة</p>
  </body>
</html>
```

القوائم Lists

توفر لغة HTML نوعين من عناصر القوائم:

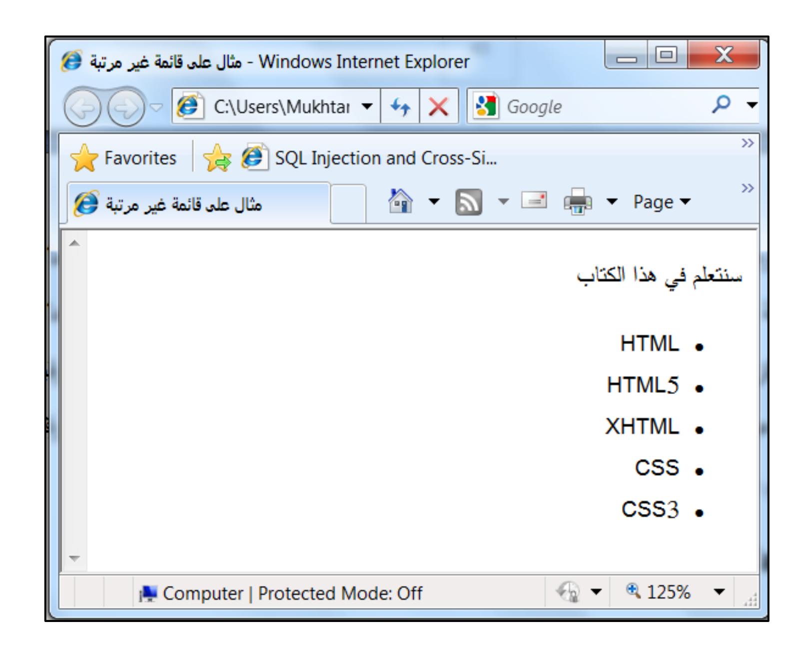
١- القوائم غير المرتبة Unordered List عبر الوسم <ul>.

٢- القوائم المرتبة Ordered List عبر الوسم <ol>.

بعد أن نقوم بتحديد نوع القائمة التي نرغب بإنشائها نضيف محتواها كمجموعة من عناصر <li> و التي تعني List Item، لنشاهد المثال التالي و الذي يعرض كيفيَّة إنشاء قائمة غير مرتبة بسيطة:

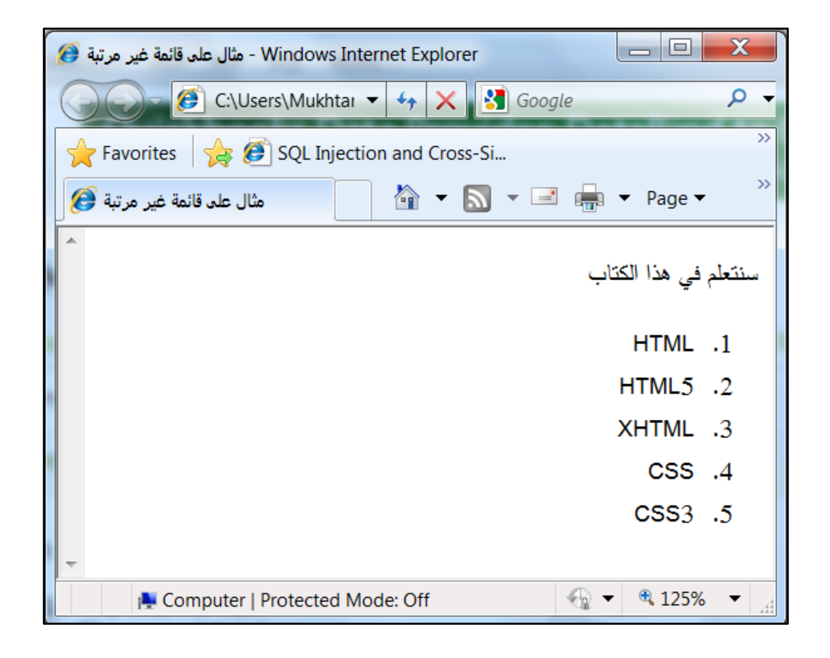
```
<html>
  <head>
     <title>مثال على قائمة غير مرتبة</title>
  </head>
  <body dir="rtl">
     <p>سنتعلم في هذا الكتاب</p>
     <ul>
          <li>HTML</li>
          <li>HTML5</li>
          <li>XHTML</li>
          <li>CSS</li>
          <li>CSS3</li>
     </ul>
  </body>
</html>
```

و الذي يبدو عند عرضه في المستعرض كمايلي:



الشكل ١٤ : مثال لقائمة غير مرتبة بسيطة

بتغيير وسمي البداية و النهاية لعنصر القائمة من <ul> و إلى <ol> و ستصبح القائمة مرتبة بالشكل التالي:



الشكل ١٥ : مثال على قائمة مرتبة بسيطة

بالطبع و كما ذكرت منذ بداية هذا الفصل فإن أي عنصر من عناصر HTML يمكن أن يحوي بين وسمي بدايته و نهايته أية مجموعة أخرى من العناصر و هذا يعني أننا نستطيع إنشاء قائمة تحوي قوائم فرعيَّة كعناصر لها، و المثال التالي يوضح هذه الفكرة:

```
<html>
  <head>
      <title>مثال على قائمة معقدة</title>
  </head>
  <body dir="rtl">
      <p>سنتعلم في هذا الكتاب</p>
      <ol>
            <li>
               HTML
               <ul>
                   <li>الروابط</li>
                   <li>الصور</li>
                   <li>القوائم
                         <ul>
                              <li>المرتبة</li>
                              <li>غير المرتبة</li>
                         </ul>
                   </li>
              </ul>
            </li>
            <li>HTML 5</li>
```

```
            <li>XHTML</li>

            <li>CSS</li>

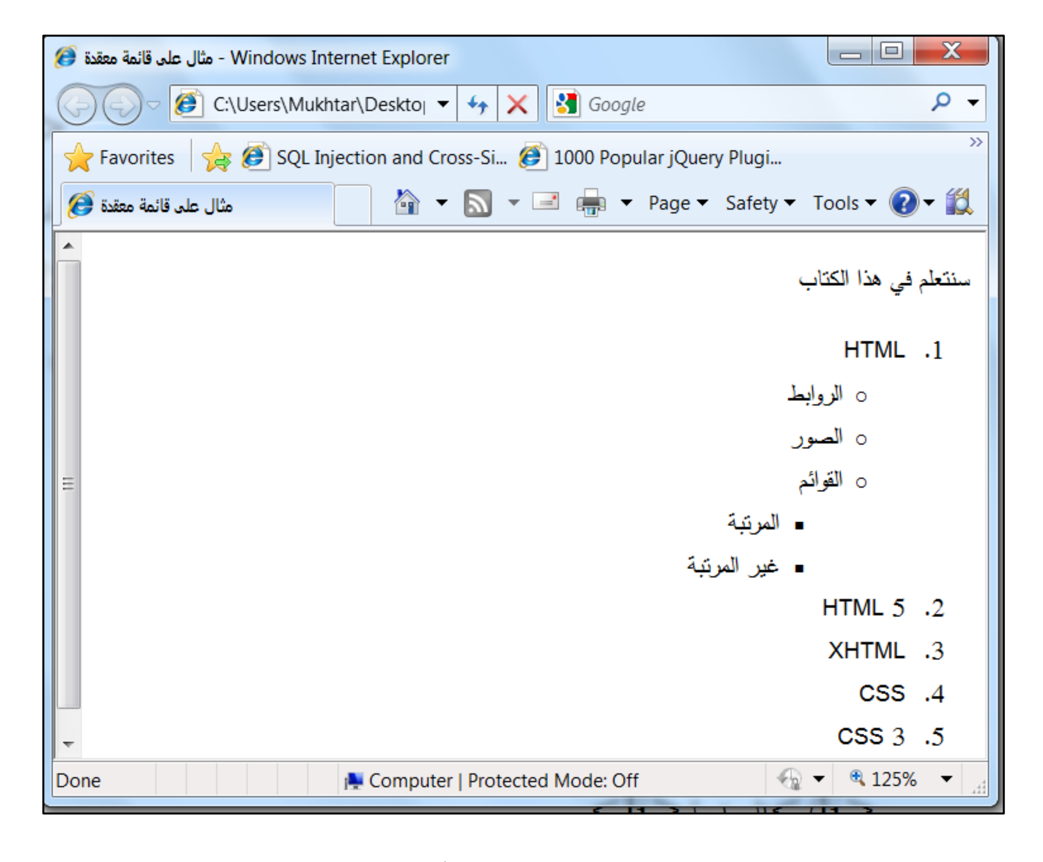
            <li>CSS 3</li>

       </ol>

   </body>

</html>
```

يبدو المثال أعلاه في المستعرض كمايلي:



الشكل ١٦ : مثال على قائمة معقدة

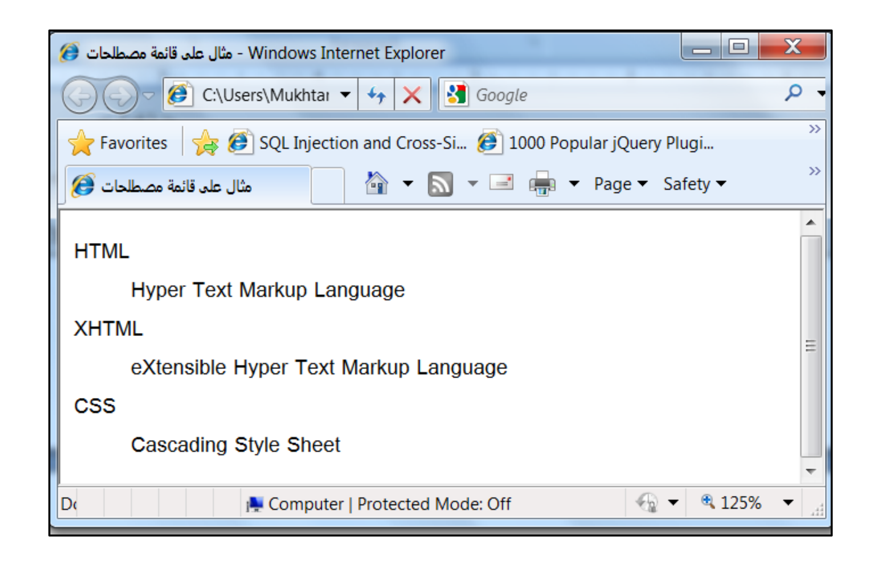
آخر ما يجب ذكره عن القوائم أنّه يمكن استخدام الواصفة type لتغيير شكل القائمة ففي حالة القوائم غير المرتبة يمكن إسناد إحدى القيم التالية إليها: square أو disc أو circle لوضع رمز المربع أو القرص أو الدائرة أمام كل عنصر على الترتيب، و في حالة القائمة المرتبة يمكن إسناد

إحدى القيم ١ أو A أو a أو I أو i إلى الواصفة type لتحويل الترقيم إلى ترقيم بأرقام عربية، ترقيم بأحرف إنجليزية كبيرة، ترقيم بأحرف إنجليزية صغيرة، ترقيم بأرقام لاتينية كبيرة، ترقيم بأرقام لاتينية صغيرة على الترتيب.

يوجد نوع خاص من القوائم يعرف بقوائم المصطلحات يتم إنشاؤه باستخدام الوسم <dl> لعنصر القائمة و الوسم <dt> لعنصر المصطلح و الوسم <dd> لعنصر تعريف المصطلح، لنشاهد المثال التالي:

```
<html>
 <head>
     <title>مثال على قائمة مصطلحات</title>
 </head>
 <body dir="rtl">
     <dl>
          <dt>HTML</dt>
          <dd>Hyper Text Markup Language</dd>
          <dt>XHTML</dt>
          <dd>eXtensible Hyper Text Markup Language</dd>
          <dt>CSS</dt>
          <dd>Cascading Style Sheet</dd>
     </dl>
 </body>
</html>
```

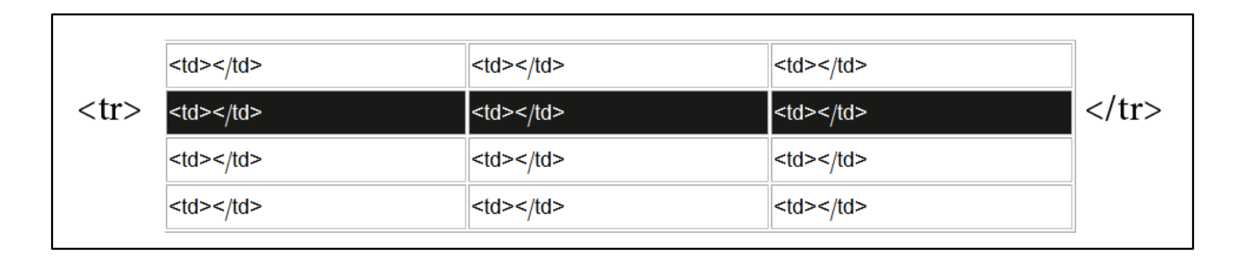
تبدو قائمة المصطلحات في المستعرض كمايلي:



الشكل ١٧ : هكذا تبدو قائمة المصطلحات في المستعرض

الجداول Tables

توفر لغة HTML آلية لإنشاء الجداول عبر العنصر <table> و يتم ذلك عبر إنشاء العنصر <table> أوّلاً و تمرير واصفتي العرض width و عرض الحدود border بالبكسل، و من ثمَّ إنشاء محتويات الجدول عبر عنصر أسطر الجدول table rows و الذي يتم تحقيقه باستخدام الوسم <tr>، ثمَّ يتم ذكر محتويات خلايا table data كل سطر من أسطر الجدول على حدى و ذلك عبر الوسم <td>، و لتسهيل حفظ الوسوم يمكن تمثيلها بشكل مرئي كمايلي:



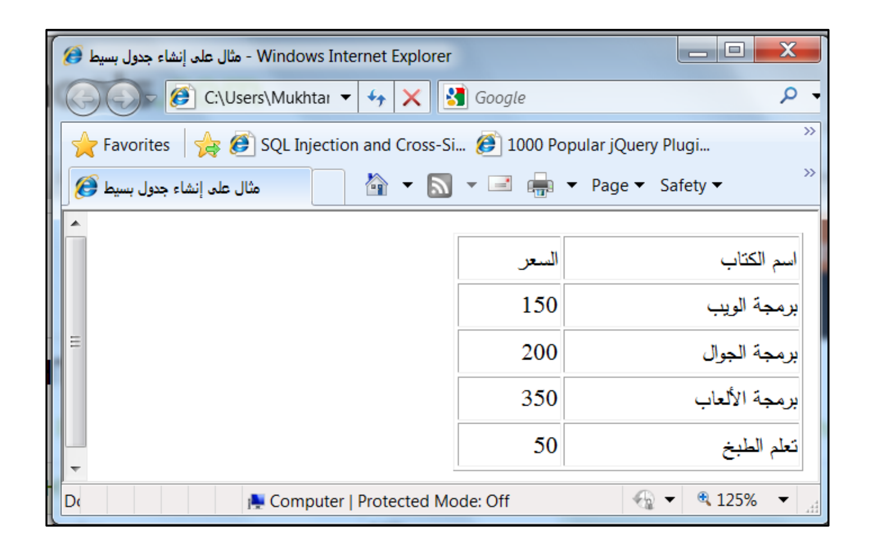
الشكل ١٨ : شكل تخيلي لتوضيح الوسوم المستعلمة لإنشاء الجدول

و لتطبيق هذا المفهوم عملياً بغية توضيحه أكثر دعنا نشاهد الشيفرة التالية التي تقوم بإنشاء جدول بسيط:

```
<html>
  <head>
      <title>مثال على إنشاء جدول بسيط</title>
  </head>
  <body dir="rtl">
      <table border="1" width="50%">
           <tr>
                <td>اسم الكتاب</td>
                <td>السعر</td>
          </tr>
           <tr>
                <td>برمجة الويب</td>
                <td>١٥٠</td>
          </tr>
           <tr>
                <td>برمجة الجوال</td>
                <td>٢٠٠</td>
          </tr>
           <tr>
                <td>برمجة الألعاب</td>
                <td>٣٥٠</td>
          </tr>
           <tr>
                <td>تعلم الطبخ</td>
                <td>٥٠</td>
```

```
            </tr>
        </table>
    </body>
</html>
```

يبدو هذا الجدول عند عرضه في مستعرض الويب كمايلي:



الشكل ١٩ : مثال على إنشاء جدول بسيط

كما رأينا فقد قمنا أولاً بإنشاء عنصر الجدول <table> بعرض width ٥٠% من العرض الكلي للصفحة (حاول تغيير حجم نافذة المستعرض لتلاحظ أن ذلك يبقى صحيحاً!)،و بعرض حدود border ١ بكسل، ثمَّ قمنا بإنشاء سطر جديد باستخدام الوسمين <tr></tr>، يحوي هذا السطر على خليتين <td></td> تحوي الأولى القيمة "اسم الكتاب" و تحوي الثانية القيمة "السعر"، ثمَّ قمنا بتكرار نفس الأسلوب مع الأسطر الأخرى التي تحوي بيانات الكتب، أظن أن الموضوع أصبح واضحاً الآن.

ملحوظة: يمكن إعطاء قيمة عرض الجدول بالبكسل أيضاً أو بأي واحدة قياس من الواحدات المدعومة في لغة HTML و التي سيتم ذكرها لاحقاً.

يقسم كل جدول من جداول HTML إلى أربع مناطق:

١- منطقة رأس الجدول Table Head: يُنشأ هذا القسم باستخدام الوسمين <thead> و </thead> و يحوي في الغالب على عنصر سطر <tr></tr> يحوي بين وسمي بدايته و نهايته عناوين أعمدة الجدول في خلايا من النوع <th> بدلاً من <td>، حيث تم الاصطلاح أن <td> تعني خلية بيانات table data في حين أن <th> تعني خلية عنوان table heading.

٢- منطقة جسم الجدول Table Body: يُنشأ هذا القسم باستخدام الوسمين <tbody> و </tbody> و يحوي على سطور البيانات و التي تنشأ باستخدام <tr> و بما أن خلاياها تحوي معلومات فإن الخلايا تنشأ باستخدام الوسوم <td>.

٣- منطقة ذيل الجدول Table Foot: يُنشأ هذا القسم باستخدام الوسمين <tfoot> و </tfoot> و يحوي في الغالب على عنصر سطر <tr></tr> يحوي بين وسمي بدايته و نهايته ملخصات عن أعمدة الجدول في خلايا من النوع <th>.

٤- عنوان الجدول Caption: يُضاف عنوان الجدول بكتابة نص العنوان بين وسمي <caption> و </caption>، يوضِّح الشكل التالي مناطق الجدول المختلفة:

عنوان الجدول ← فاتورة شراء الكتب

اسم الكتاب	السعر
برمجة الويب	150
برمجة الجوال	200
برمجة الألعاب	350
تعلم الطبخ	50
المجموع	**750**

رأس الجدول ←

جسم الجدول ←

ذيل الجدول ←

الشكل ٢٠ : رسم توضيحي لأقسام الجدول في HTML

لتحقيق الجدول المذكور في صفحتنا سنكتب شيفرة HTML التالية:

```
<html>
  <head>
      <title>مثال على إنشاء جدول مثالي</title>
  </head>
  <body dir="rtl">
      <table border="1" width="50%">

        <caption>فاتورة شراء الكتب</caption>

        <thead>
             <tr>
                  <th>اسم الكتاب</th>
                  <th>السعر</th>
             </tr>
        </thead>

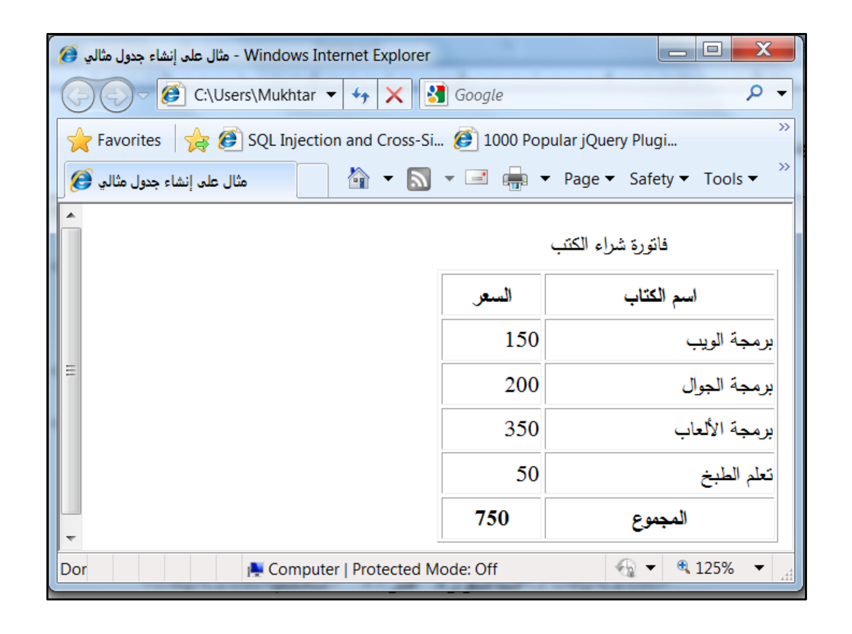
        <tbody>
             <tr>
                     <td>برمجة الويب</td>
                     <td>150</td>
            </tr>
             <tr>
                     <td>برمجة الجوال</td>
```

```
                    <td>200</td>
            </tr>
            <tr>
                    <td>برمجة الألعاب</td>
                    <td>350</td>
            </tr>
            <tr>
                    <td>تعلم الطبخ</td>
                    <td>50</td>
            </tr>
        </tbody>

        <tfoot>
            <tr>
                <th>المجموع</th>
                <th>750</th>
            </tr>
        </tfoot>

    </table>
  </body>
</html>
```

تبدو هذه الشيفرة في مستعرض الويب كمايلي:



فاتورة شراء الكتب

اسم الكتاب	السعر
برمجة الويب	150
برمجة الجوال	200
برمجة الألعاب	350
تعلم الطبخ	50
المجموع	**750**

الشكل ٢١ : مثال لإنشاء جدول HTML مثالي

كما تلاحظ فإنَّ العنوان يظهر أعلى الجدول، كما أن الأسطر الموجودة في منطقتي الرأس و الذيل تظهر بخط عريض bold.

الجداول غير البسيطة

في الحقيقة ما تم استعراضه في الفقرة الماضية كان جميلاً و بسيطاً أيضاً فبالنسبة لحالة مثل حالة فاتورة الشراء فإن جدولاً بعمودين و بمجموعة من الأسطر أمر بسيط و لكن ماذا عن الحالات الأعقد؟، ماذا عن الحالات التي لا تكون فيها الخلايا متساوية الأحجام؟، و لا الأسطر متساوية عدد الخلايا؟، تجيب HTML على هذه التساؤلات المحقّة عبر تقديم الواصفتين rowspan و colspan.

تستخدم الواصفتان rowspan و colspan مع وسم بداية الخلية <td> أو <th> و تسند إليهما قيمة رقمية صحيحة (١ أو ٢ أو ٣ .. إلخ) تغير من حجم الخلية فتجعلها بحجم خليتين أو ثلاث أو أربع .. إلخ، حيث أن rowspan تتحكم بعرض الخلية، أما colspan فتتحكم بارتفاع الخلية.

لنشاهد الشيفرة التالية على سبيل المثال:

```
<html>
  <head>
     <title>مثال على إنشاء جدول غير بسيط</title>
  </head>
  <body dir="rtl">
     <table border="1" width="100%">
        <tbody>
             <tr>
               <td>اسم الكتاب</td>
               <td>تعلم jQuery في ١٢٠ دقيقة</td>
               <td rowspan="3">
                 <img src="jQuery120.png">
               </td>
            </tr>
             <tr>
              <td>المؤلف</td>
               <td>مختار سيد صالح</td>
            </tr>
             <tr>
                <td>سنة الإصدار</td>
                <td>2010</td>
            </tr>
            <tr>
               <td colspan="3">
يقدم هذا الكتاب معلومات عن مكتبة jQuery
```

```
بشكل مبسط و واضح ..... إلخ.
                </td>
            </tr>
        </tbody>
    </table>
  </body>
</html>
```

تبدو الشيفرة السابقة عند عرضها في المستعرض كمايلي:



الشكل ٢٢ : مثال على إنشاء جدول غير بسيط باستخدام الواصفتين rowspan و colspan

كما ترى فإنّ الجدول يتكون من أربعة سطور يحوي السطر الأوَّل على ثلاث خلايا واحدة منها بارتفاع يعادل ارتفاع ثلاثة أسطر مجتمعةً (هذا ما تعنيه rowspan="3")، أمَّا السطر الأخير من الجدول فيحوي على خلية واحدة بعرض أعمدة الجدول الثلاثة مجتمعةً (و هذا ما تعنيه

"colspan="3)، أظن أن استخدام الواصفتين rowspan و colspan أصبح واضحاً الآن، و بالنسبة لأصدقائي الذين لم تتضح الصورة بالنسبة لهم فأعتذر منهم و أنصحهم أن يقوموا بتغيير القيم المذكورة في الواصفتين rowspan و colspan في المثال السابق و يلاحظوا الفرق الحاصل في أحجام الخلايا فهذا يساعد كثيراً في تشكيل تصوّر أفضل لديهم كما أظن.

ملحوظة: ليطمئن قلبي أحب أن أذكرك أنه بإمكانك وضع أي من عناصر HTML كمحتوى في خلايا الجدول.

النماذج Forms

تستخدم النماذج Forms في لغة HTML لاستقبال المدخلات Inputs من المستخدم بغية عرضها أو تخزينها أو القيام بعمليات معالجة معينة عليها، يتم إنشاء النماذج باستخدام الوسمين <form> و </form> الذان سيحويان بينهما مجموعة من عناصر الإدخال <input> مثل مربعات النصوص text boxes و عناصر الاختيار check boxes و الأزرار buttons و القوائم المنسدلة select lists.. إلخ.

يتم تزويد وسم بداية عنصر النموذج <form> بواصفتين أساسيتين الأولى هي الواصفة action و التي تحدد اسم الملف البرمجي الذي سيقوم بمعالجة البيانات المدخلة في النموذج و يكون الملف البرمجي مكتوباً بإحدى لغات البرمجة من طرف السيرفر Server Side مثل PHP أو ASP.NET أو JSP أو .. إلخ (و البرمجة من طرف السيرفر موضوع خارج نطاق هذا الكتاب)، أمّا الواصفة الثانية فهي الواصفة method و التي تحدد طريقة إرسال البيانات إلى الملف البرمجي المذكور سابقاً بإحدى طريقتين:

١ – الطريقة الأولى هي الطريقة get حيث يتم إرسال البيانات في شريط العنوان في المستعرض بعد اسم الملف البرمجي بشكل أزواج مفتاح/قيمة، انظر للرابط التالي على سبيل المثال:

http://www.site.com/file.php?name=Mukhtar&age=23&job=ComputerEngineer

كما تلاحظ فإن الملف البرمجي في هذه الحالة هو http://www.site.com/file.php أمَّا القيم المرسلة فهي:

name = Mukhtar و age = 23 و job = ComputerEngineer

٢ – الطريقة الثانية هي الطريقة post حيث يتم إرسال البيانات بشكل غير ظاهر في شريط العنوان (في الحقيقة يتم إرسالها مع ترويسة طلب HTTP).

بما أننا –في هذا الكتاب– سنهتم بكيفية إنشاء النماذج فقط و لن نناقش كيفية معالجة الطلبات فبإمكانك الآن تجاهل ما تم ذكره عن الواصفة method و التركيز على كيفية بناء النماذج.

بعد إنشاء العنصر <form> سنقوم بوضع مجموعة من عناصر الإدخال بين وسمي بدايته و نهايته باستخدام الوسم <input> و طبعاً سيتم تزويد كل عنصر إدخال باسم فريد يُسنَد كقيمة للواصفتين name و id، و من ثمَّ سيتم تحديد نوع عنصر الإدخال عبر الواصفة type و الجدول التالي يوضح القيم الممكنة و ما يمثله كل منها:

القيمة	ما تمثله
button	زر أوامر
checkbox	عنصر اختيار متعدد
file	مربع لاختيار ملف بغية رفعه إلى الموقع
hidden	عنصر إدخال مخفي
password	عنصر إدخال كلمة المرور
radio	عنصر اختيار فردي
reset	زر لإعادة ضبط القيم الافتراضية لعناصر

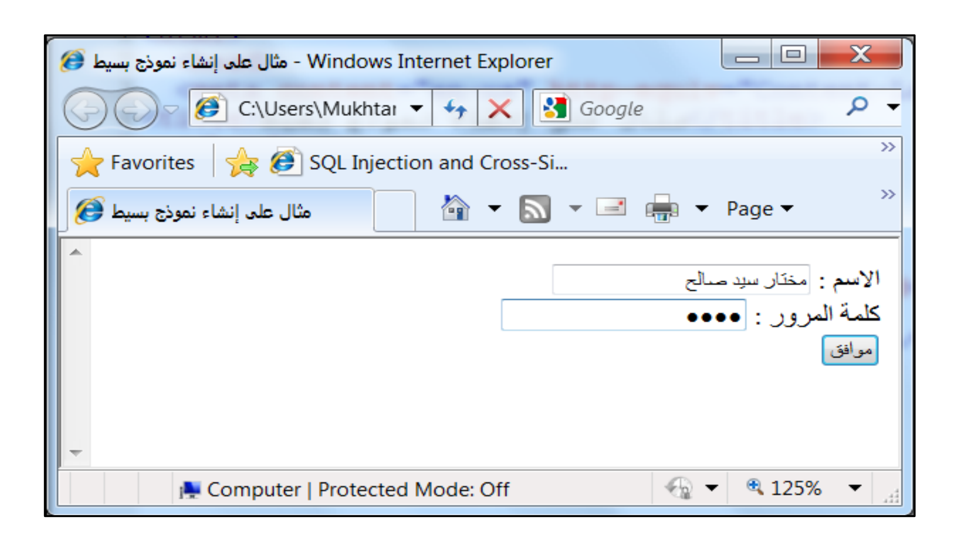
	الإدخال كاملةً في النموذج
submit	زر موافق (إرسال الطلب)
text	مربع إدخال نص

الجدول ٣ : القيم المختلفة للواصفة type الخاصة بعنصر <input>

لا تقلق إنْ بدت لك المعلومات كثيرة و معقدة و تابع معي الشيفرة التالية:

```
<html>
  <head>
     <title>
        مثال على إنشاء نموذج بسيط
     </title>
  </head>
  <body dir="rtl">
     <form method="get" action="file.php">
        الاسم :
      <input type="text" name="nameText" id="nameText" />
      <br/>
        كلمة المرور :
      <input type="password" name="pass" id="pass" />
      <br/>
       <input type="submit" name="okButton" id="okButton"
value="موافق" />
     </form>
  </body>
</html>
```

و التي تبدو عند عرضها في مستعرض الويب و بعد إدخال بعض البيانات كمايلي:



الشكل ٢٣ : مثال على إنشاء نموذج بسيط

حسناً الموضوع بسيط كما رأيت، دعنا نقوم بتطوير نموذجنا هذا ليشمل مجموعة أخرى من عناصر الإدخال على سبيل التجربة، انظر للشيفرة التالية:

```
<html>
  <head>
     <title>مثال على إنشاء نموذج أعقد قليلاً</title>
  </head>
  <body dir="rtl">
     <form method="get" action="file.php">
          الاسم: <input type="text" name="nameText" id="nameText" />
          <br/>
          كلمة المرور: <input type="password" name="pass" id="pass" />
          <br/>
```

```
            الصورة الشخصية: <input type="file" name="pic"
id="pic">

            <br/>

            <input type="reset" name="resBtn" id="resBtn"
value="استعادة" />

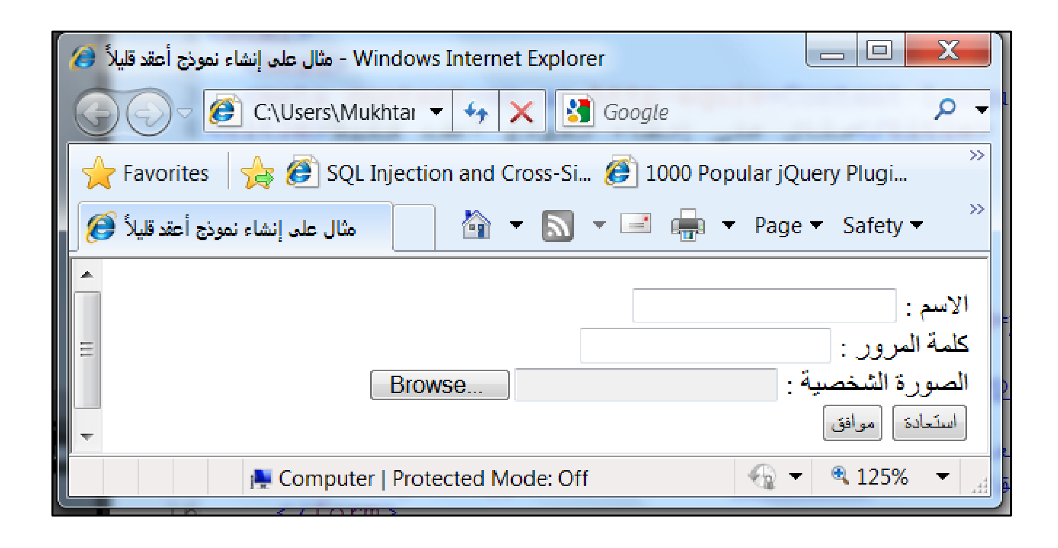
            <input type="submit" name="okButton" id="okButton"
value="موافق" />

      </form>

  </body>

</html>
```

تبدو الشيفرة السابقة في مستعرض الويب كمايلي:



الشكل ٢٤ : إنشاء نموذج أعقد بقليل

مهلاً! .. و لكن ماذا عن عناصر الاختيار radio و checkbox ؟

حسناً، في الواقع يتم استعمال هذه العناصر في النماذج بغية تمكين المستخدم من الإجابة على سؤال محدد و ذلك باختيار جواب واحد فقط من عدة إجابات محتملة (في حالة عنصر radio) أو الإجابة على سؤال محدد باختيار أكثر من جواب (في حالة عنصر checkbox) و يتم استخدام

هذه العناصر بنفس الطريقة السابقة مع ملحوظة إعطاء جميع العناصر التي تمثل إجابة لنفس السؤال القيمة ذاتها للواصفة name و قيماً مختلفة للواصفة value، لنشاهد المثال التالي:

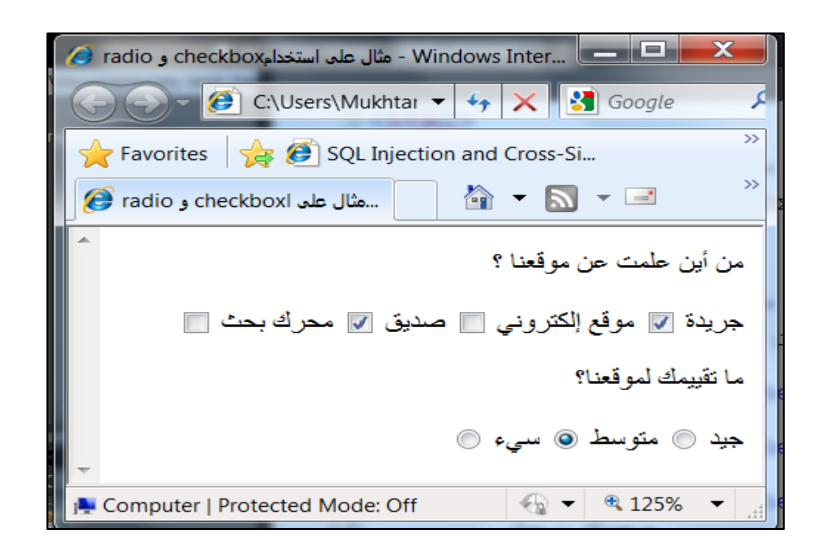
```
<html>
  <head>
      <title> radio و checkbox مثال على استخدام</title>
  </head>
  <body dir="rtl">
    <form method="post" action="file.php">
     <p>من أين علمت عن موقعنا ؟</p>
     جريدة
     <input type="checkbox" name="wrKnow" value="nPaper" />
     موقع إلكتروني
     <input type="checkbox" name="wrKnow" value="wbSite" />
     صديق
     <input type="checkbox" name="wrKnow" value="frnd" />
     محرك بحث
     <input type="checkbox" name="wrKnow" value="srEng" />
     <p>ما تقييمك لموقعنا؟</p>
     جيد
     <input type="radio" name="rate" value="good" />
     متوسط
     <input type="radio" name="rate" value="mid" />
     سيء
     <input type="radio" name="rate" value="bad" />
```

```
</form>

  </body>

</html>
```

و الذي يبدو في المستعرض كمايلي:



الشكل ٢٥ : مثال على استخدام **checkbox** و **radio**

هناك طريقة أخرى لتحقيق شيء مشابه لعنصر radio من حيث الهدف و لكنه يستخدم في حال وجود بدائل كثيرة (أجوبة كثيرة محتملة للسؤال) و هو عنصر القائمة المنسدلة و يتم تحقيقه عبر الوسم <select> أمَّا عناصر القائمة فيتم تحقيقها من خلال الوسم <option> و يتم تحديد أحد هذه العناصر كخيار افتراضي عن طريق تمرير الواصفة "selected="selected في وسم بدايته، لنشاهد المثال التالي:

```
<html>

  <head>

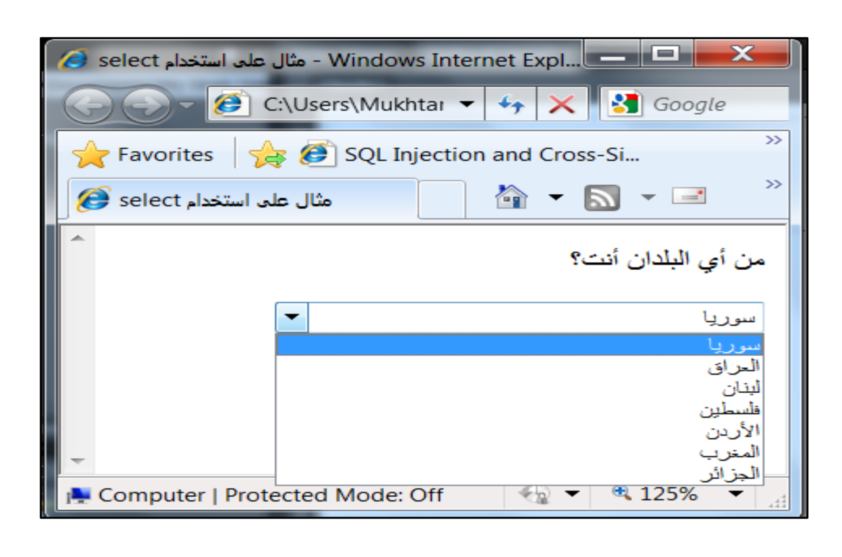
      <title>select مثال على استخدام</title>

  </head>

  <body dir="rtl">
```

```
      <form method="post" action="file.php">
         <p>من أي البلدان أنت؟</p>
         <select name="nationality">
           <option selected="selected">سوريا</option>
           <option>العراق</option>
           <option>لبنان</option>
           <option>فلسطين</option>
           <option>الأردن</option>
           <option>المغرب</option>
           <option>الجزائر</option>
         </select>
      </form>
   </body>
</html>
```

و الذي يبدو في المستعرض كمايلي:



الشكل ٢٦ : مثال على استخدام عنصر select

بقي عنصر إدخال أخير لمناقشته هنا و هو العنصر <textarea> و الذي يستخدم لإدخال قيمة نصية متعددة الأسطر على عكس العنصر <input> من النوع text و الذي يستخدم لإدخال قيمة نصية وحيدة السطر.

يمرر للعنصر <textarea> قيم عددية صحيحة عبر الواصفتين cols و rows لتحديد عرضه و ارتفاعه، انظر الشيفرة التالية:

```
<html>
  <head>
      <title>
           مثال على استخدام textarea
     </title>
  </head>
  <body dir="rtl">
     <form method="post" action="file.php">
        <p>اكتب نبذة قصيرة عنك</p>

        <textarea cols="30" rows="10" name="bio">
        </textarea>


    </form>
  </body>
</html>
```

و الذي يبدو عند استعراضه في مستعرض الويب كمايلي:



الشكل ٢٧ : مثال على استخدام العنصر textarea

تقسيم عناصر الإدخال إلى مجموعات

يمكن تقسيم عناصر الإدخال في النماذج الكبيرة إلى أقسام منطقية يوضع كل منها في إطار مستقل و هذا ما توفره لغة HTML باستخدام الوسمين <fieldset> و </fieldset> اللذان سيحويان بينهما مجموعة عناصر الإدخال و يتم تزويد كل مجموعة بعنوان توضيحي عبر الوسم <legend>، كما يمكن منح كل عنصر من عناصر الإدخال عنواناً يدل على ماهيَّة المعلومات المدخلة فيه عن طريق الوسم <label> بتمرير اسم عنصر الإدخال إلى واصفة for الخاصة بعنصر العنوان <label>، دعنا نشاهد المثال التالي:

```
<html>
  <head>
      <title>
          مثال على استخدام fieldset
     </title>
  </head>
```

```
  <body dir="rtl">
      <form method="get" action="">

           <fieldset>
                <legend>البيانات الشخصية</legend>
                <label for="nameTXT">الاسم</label>
                <input type="text" name="nameTXT" id="nameTXT"
maxlength="50" />
                <br/><br/>
                <label for="ageTXT">العمر</label>
                <input type="text" name="ageTXT" id="ageTXT"
maxlength="2" />
           </fieldset>

           <br/>

           <fieldset>
                <legend>مهارات العمل</legend>
                C#
                <input type="checkbox" name="prog" value="cs"/>
                VB.NET
                <input type="checkbox" name="prog" value="vb"/>
                F#
                <input type="checkbox" name="prog" value="fs"/>
                C++.NET
                <input type="checkbox" name="prog" value="cpp"/>
```

```
            </fieldset>

            <fieldset>

                  <input type="submit" value="إرسال" />

            </fieldset>

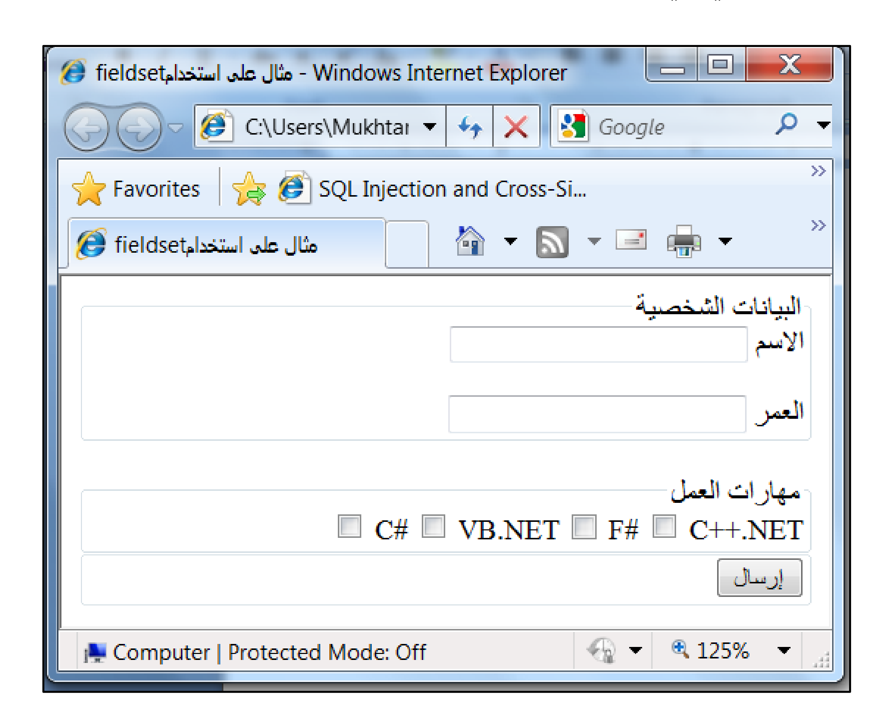


      </form>

  </body>

</html>
```

تبدو الشيفرة السابقة كمايلي في مستعرض الويب:



الشكل ٢٨ : مثال على استخدام fieldSet

الإطارات Frames

تسمح لك لغة HTML بتضمين صفحات ويب ضمن صفحات أخرى، و ذلك عبر استخدام الإطارات Frames.

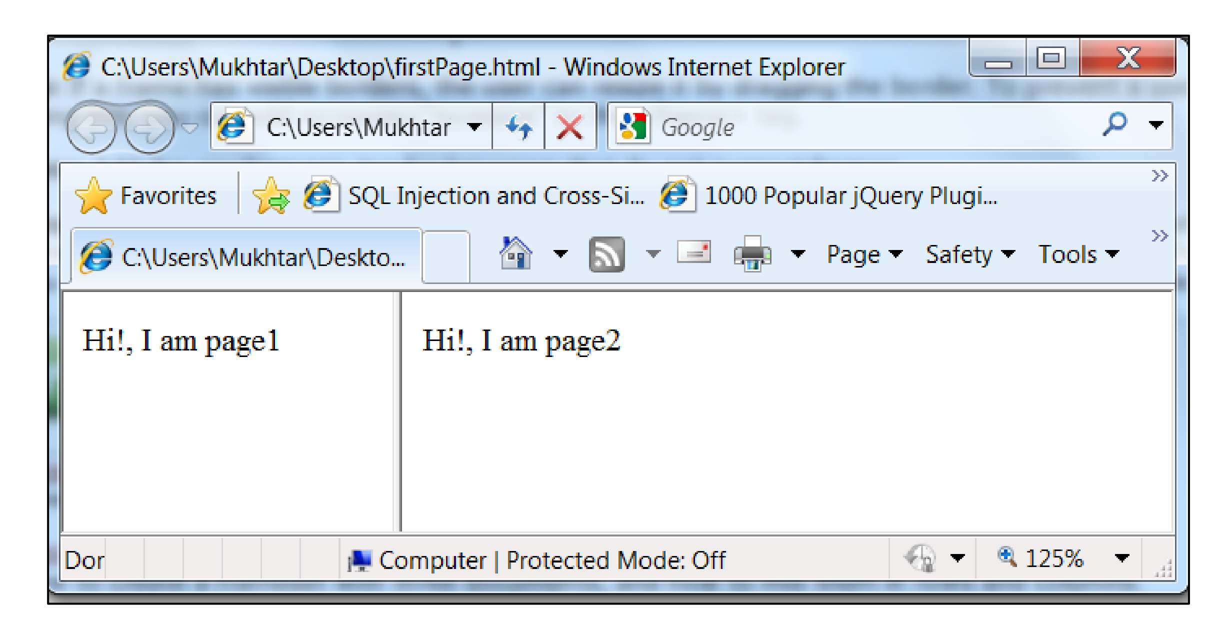
يتم استعمال الإطارات في الصفحة على مرحلتين، الأولى بإنشاء حاوية الإطارات باستخدام الوسم <frameset> و الذي يحدد بدايةً عدد الإطارات التي سيتم تضمينها، و الحجم الذي سيأخذه كل إطار من حجم الصفحة الكلي، و يتم تحديد الحجم بالبكسل أو بالنسبة المئوية، أمّا المرحلة الثانية فيتم فيها إضافة الإطارات بين وسمي البداية و النهاية للعنصر <frameset> و ذلك باستخدام الوسم <frame> و إسناد مسار الصفحة التي سيحويها الإطار للواصفة src ، لنشاهد المثال التالي لصفحة ويب تضم إطارين الأوّل بحجم قدره ٣٠% من حجم الصفحة الكلي و الثاني بالحجم الباقي من الصفحة:

```
<html>

     <frameset cols="30%,*">
          <frame src="page_1.htm" />
          <frame src="page_2.htm" />
     </frameset>

</html>
```

يبدو هذا في مستعرض الويب كمايلي:

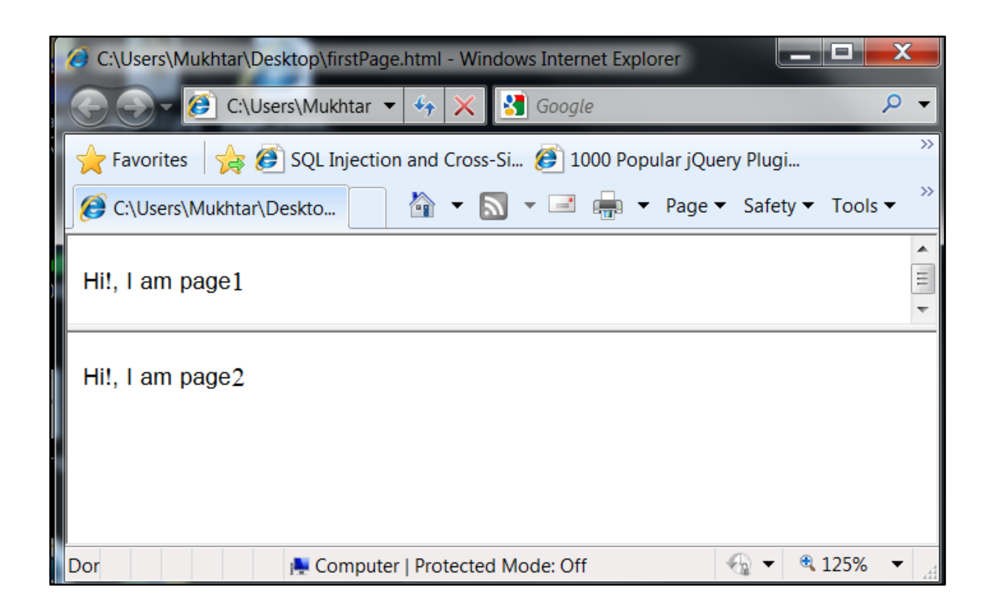


الشكل ٢٩ : استخدام الإطارات بشكل عمودي

ملحوظة: الصفحة التي تستعمل فيها الإطارات لا تحوي على وسم <body>.

يمكن أن يقوم المستخدم بتغيير حجم الإطارات عن طريق تحريك الخط الفاصل بينهما و الظاهر بوضوح في الشكل السابق (انظر الشكل ٢٩)، يمكن إلغاء هذا السلوك للإطارات عن طريق إضافة الواصفة "noresize="noresize إلى وسوم <frame>.

في المثال السابق كانت الإطارات بشكل عمودي و لهذا تم استعمال الواصفة cols مع الوسم <frameset>، و يمكن جعل الإطارات بشكل أفقي عن طريق استبدال الواصفة cols بالواصفة rows، لتصبح الصفحة كمايلي في مستعرض الويب:



الشكل ٣٠ : استخدام الإطارات بشكل أفقي

تذكرة: يمكن جعل الرابط يعرض محتواه في إطار معيَّن عن طريق إسناد اسم الإطار إلى واصفة target الخاصة بالرابط <a> (انظر فقرة الروابط التشعبيَّة Links).

في حين أن الوسمين <frameset> و <frame> يستخدمان لإنشاء صفحة أم تضم مجموعة من الصفحات <u>فقط</u> في إطاراتها فإن لغة HTML توفر الوسم <iframe> لتضمين صفحة ويب في

صفحة أخرى بشكل متناسق مع بقية محتوى الصفحة، و يستخدم الوسم <iframe> بالصيغة التالية:

```
<iframe src="URL"></iframe>
```

حيث أنَّ URL هو مسار الصفحة التي سيتم تضمينها، و يمكن ضبط حجم الإطار عن طريق الواصفتين height و width بالبكسل أو بالنسبة المئوية.

آخر ما يجب ذكره هنا أن هناك مستعرضات لا تدعم الإطارات، كما أنّ الإطارات ستلغى مستقبلاً من اللغة، على الرغم من أن بعض المستعرضات ما زالت تدعمها إلى تاريخ كتابة هذه الأسطر.

عناصر الـ Meta

ذكرت في بداية هذا الفصل أنَّ هناك عناصر غير مرئية تضاف في منطقة رأس الصفحة بين وسمي <head> و </head> تستخدمها محركات البحث من أجل تصنيف صفحتك و عرضها ضمن نتائج البحث، و هذه العناصر هي عناصر الـ meta، و بالطبع فمن الأهمية بمكان جعل محركات البحث تصنّف صفحتك بشكل صحيح كي لا تخسر معظم زوار موقعك القادمين إثر عملية بحث ما على أحد محركات البحث.

يتم تحقيق عناصر الـ meta من خلال الوسم meta و بتمرير اسم name و محتوى content لعنصر الـ meta كواصفتين له، و لعلَّ أهم عنصري meta في أي صفحة ويب هما العنصران اللذان يحملان الاسمين keywords و description، و اللذان يستخدمان كمايلي:

```
<html>

      <head>

           <meta name="keywords" content="xhtml website test
meta head"/>

           <meta name="description" content="this page
demonstrates how to use meta elements"/>
```

```
    </head>
    <body>محتوى الصفحة النهائي فعليّاً سيكون هنا</body>
</html>
```

حيث يعرض عنصر الـ Meta ذو الاسم Keywords مجموعة من الكلمات المفتاحية التي تظن أن المستخدمين سيبحثون عن أحدها ضمن المستعرض و الموجودة في صفحتك، أما عنصر الـ Meta ذو الاسم Description فيحوي وصفاً مختصراً عن المحتوى الذي تقدمه صفحتك.

بالطبع عند عرض هذه الصفحة في مستعرض الويب لا تتوقع أن تشاهد محتوى عناصر الـ Meta ظاهراً فمهمّة هذه العناصر كما قلنا هو تقديم بيانات عن البيانات الموجودة ضمن صفحتك.

و بانتهاء حديثنا عن عناصر الـ Meta نكون قد أنهينا –بفضل الله– مناقشة لغة HTML بأغلب وسومها، و سنتابع في الصفحات التالية استعراض وسوم HTML العصريَّة و التي تعرف باسم HTML5 ، و لكن قبل ذلك سنستعرض لغة HTML الموسَّعة أو ما يُعرف بـ XHTML، أمَّا الآن فأرى أن نعرض جدولاً بوسوم HTML التقليديَّة ليكون مرجعاً سريعاً عند الحاجة.

دليل وسوم HTML

الوسم	الشرح
<!-- ... -->	تعليق
<!DOCTYPE>	لتحديد نوع المستند (سيتم الحديث عنه عند عرض XHTML)
<a>	رابط تشعبي
<abbr>	اختصار لمصطلح إنجليزي مركب
<acronym>	اختصار لمصطلح إنجليزي
<address>	إضافة معلومات اتصال بمالك الصفحة
<applet>	تضمين تطبيق Applet ضمن الصفحة
<area />	تعريف منطقة ضمن الخريطة الصوريَّة
<b>	نص عريض
<base />	تعريف بادئة افتراضية لمسارات الروابط
<basefont />	تعريف نوع/لون/حجم خط افتراضي لنصوص للصفحة
<bdo>	تعريف اتجاه النص
<big>	نص كبير الحجم
<blockqoute>	اقتباس طويل
<body>	جسد الصفحة
 	سطر جديد
<button>	زر إرسال
<caption>	عنوان الجدول
<center>	لتوسيط المحتوى
<cite>	اقتباس متوسط
<code>	شيفرة برمجية
<col />	تعريف واصفات مشتركة لأعمدة الجدول
<colgroup>	تعريف مجموعة من أعمدة الجدول لتطبيق واصفات مشتركة عليها

<dd>	وصف لمصطلح ضمن قائمة المصطلحات
<del>	نص مشطوب
<dfn>	تعريف مصطلح ضمن قائمة المصطلحات
<dir>	قائمة مجلدات
<div>	قسم من المستند
<dl>	قائمة المصطلحات
<dt>	عنصر قائمة خاص بقائمة المصطلحات
<em>	نص مشدد عليه
<fieldset>	مجموعة حقول في نموذج
<font>	تعريف حجم/نوع/لون الخط لنص معين
<form>	نموذج
<frame />	إطار
<frameset>	مجموعة إطارات
<h1> إلى <h6>	عناصر العناوين
<head>	رأس المستند
<hr />	خط أفقي
<html>	عنصر بداية المستند
<i>	مائل
<iframe>	إطار كعنصر من عناصر الصفحة
<img />	صورة
<input />	عنصر إدخال
<ins>	نص تحته خط
<isindex>	إنشاء فهرس قابل للبحث يتعلق بالصفحة
<kbd>	نص بخط حاسوبي برمجي
<label>	عنوان لعنصر ما

<legeng>	عنوان لمجموعة عناصر الإدخال <fieldset>
** **	عنصر القائمة
<link />	ربط المستند بمستند آخر (سيمر معنا أحد استعمالاته في الفصل الثاني)
<map>	تعريف خريطة صورية
<menu>	تعريف قائمة
<meta />	بيانات تعريفية عن ماهيَّة محتوى المستند
<noframes>	لإنشاء بديل لمحتوى الإطار في المستعرضات التي لا تدعم الإطارات
<noscript>	لإنشار بديل لمحتوى السكربت في المستعرضات التي لا تدعم السكربت
<object>	تضمين كائن برمجي
** **	قائمة مرتبة
<optgroup>	مجموعة من الخيارات المرتبطة منطقياً ضمن خيارات القائمة المنسدلة
<option>	خيار في قائمة منسدلة
<p>	نص
<param />	وسيط للكائن البرمجي
<pre>	نص منسق مسبقاً
<q>	اقتباس قصير
<s>	نص مشطوب
<samp>	نموذج لشيفرة برمجية
<script>	سكربت برمجي (Java Script أو jQuery غالباً)
<select>	قائمة منسدلة
<small>	نص صغير الحجم
<span>	تعريف قسم سطري من المستند
<strike>	نص مشطوب
<strong>	نص عريض
<style>	تعريف ورقة أنماط (يناقش في الفصل القادم بالتفصيل)

<sub>	نص سفلي
<sup>	نص علوي
<table>	جدول
<tbody>	جسد الجدول
<td>	خلية بيانات جدول
<textarea>	عنصر إدخال نص متعدد الأسطر
<tfoot>	ذيل الجدول
<th>	خلية عنوان رأس الجدول
<thead>	رأس الجدول
<title>	عنوان المستند
<tr>	سطر الجدول
<tt>	نص teletype
<u>	نص تحته خط
	قائمة غير مرتبة
<var>	جزء متغير من النص
<xmp>	نص بتنسيق مسبق

الجدول ٤ : دليل وسوم HTML

XHTML

إنَّ XHTML هي اختصار لـ eXtensible HTML و التي تعني HTML القابلة للتوسُّع (البعض يدعوها HTML الموسَّعة)، و هي لا تختلف عن HTML التقليديَّة من ناحية الوسوم و لا من ناحية الواصفات، و لكن ببساطة يمكن اعتبارها معايير أكثر صرامةً لكتابة مستندات HTML بشكل يتوافق و قواعد لغة XML (eXtensible Markup Language)، و هذا ما يعطي صفحاتنا توافقيَّة أكبر مع مستعرضات الويب، كما يدل الالتزام بمعايير XHTML على احترافية أكبر في العمل، و بالطبع لا أريد لتفكيرنا –كمطوِّري ويب– أن يبقى محصوراً في مستعرض الويب الخاص بالحاسوب، فلنفكِّر في مستعرضات الويب الخاصة بالهواتف الذكيَّة أو بالأجهزة الصغيرة الأخرى و لنسأل أنفسنا: هل ستبدو صفحاتنا كما نريد على كل مستعرضات الويب؟، هذا ما تحاول XHTML الإجابة عنه.

ففي حين أن شبكة الانترنت تحوي مليارات صفحات الويب فثَمَّةَ نسبة كبيرة من هذه الصفحات مكتوبة بشكل رديء – للأسف – يشبه مايلي على سبيل المثال:

```
<hTml>
      <heaD>
            <title> صفحة HTML رديئة</title>
      <BODY>
            <h1>Bad HTML
            <p>This is a paragraph
      </boDy>
```

مع أن الشيفرة أعلاه ستعمل على بعض المستعرضات إلا أنها باعتقادي لن تؤمِّن أدنى فرصة عمل لمن قام بكتابتها لأنها لا تعكس أدنى معرفه بكتابة HTML و أفضل ما يمكن أن يقال عنها أنها مكتوبة بشكل رديء، و قد ظهرت XHTML للحد من انتشار صفحات كالصفحة أعلاه.

قواعد XHTML

تتلخص XHTML في مجموعة من القواعد البسيطة التي ما إنْ يتم تطبيقها حين كتابة أي مستند HTML فسيُعَدُّ المستند مستند XHTML، و أستطيع تلخيص قواعد XHTML في مجموعة النصائح التالية[٥]:

- لا تنسَ وسم الإغلاق.
- اكتب وسوم و واصفات صفحتك بأحرف صغيرة Lower Case.
- راعِ أن يكون ترتيب وسوم الإغلاق متناظراً مع وسوم الفتح.
- ضع القيم المُسنَدة للواصفات بين علامات اقتباس.
- لا تستغنِ عن أي من وسوم <html> و <body> و <head>.
- ضع تعريف نوع المستند DOCTYPE في بداية صفحتك (سيعرض في الفقرة التالية).

تعريف نوع المستند <!DOCTYPE>

تنص معايير XHTML على أن كل صفحات الويب يجب أن تحوي في أول سطر منها على سطر التعريف <!DOCTYPE> و الذي يعني تعريف نوع المستند Document Type Definition أو DTD اختصاراً، و ذلك أن مستعرضات الويب تستخدم هذا السطر كمرجعيَّة قواعديَّة لصيغة الكتابة و لأنواع الوسوم في الصفحة، إضافة إلى استخدامها الواصفة xmlns الخاصة بوسم البداية <html> و التي تصف فضاء أسماء Name Space الصفحة.

تعرض الشيفرة التالية عنصر تعريف نوع المستند الذي تصادفه في أغلب صفحات XHTML و الذي يجب أن تكتبه كأول سطر من صفحتك إضافةً للواصفة xmlns، و بهذا يصبح الشكل العام لمستند XHTML كمايلي:

---

[٥] إنَّ أغلب ما سيتم ذكره من القواعد كان قد مر معنا في الصفحات السابقة على شكل ملحوظات.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

    <head>محتوى رأس الصفحة هنا</head>

    <body>محتوى جسد الصفحة هنا</body>

</html>
```

يعتبر النوع <!DOCTYPE> المعروض أعلاه أحد أنواع المستند و يسمَّى النوع Transitional و الفقرة التالية تناقش الاختلاف بينه و بين الأنواع الأخرى.

ملحوظة: <!DOCTYPE> ليس وسماً من وسوم اللغة و إنَّما هو سطر تعريف المستند.

الأنواع المختلفة للمستند

في XHTML هناك أربعة أنواع مختلفة للمستند و هذه الأنواع هي:

١. XHTML 1.0 Strict:

يحوي هذا النوع جميع وسوم HTML عدا الوسوم <applet> و <basefont> و <center> و <dir> و <font> و <iframe> و <isindex> و <menu> و <noframes> و <s> و <strike> و <u>، و لا يسمح هذا النوع باستخدام إطارات <frameset>، و سطر تعريف المستند من هذا النوع هو:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

٢. XHTML 1.0 Tansitional:

يحوي هذا النوع جميع وسوم HTML و لكنَّه لا يسمح كسابقه باستخدام إطارات <frameset>، و سطر تعريف المستند من هذا النوع هو:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

٣. XHTML 1.0 Frameset:

يحوي جميع وسوم HTML و يسمح باستخدام الإطارات، و سطر تعريف المستند من هذا النوع هو:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

٤. XHTML 1.1:

يماثل النوع Strict و لكنّه يتيح لك إضافات وحدات برمجيَّة خاصة (وحدات Ruby مثلاً) على عكس المذكور، و سطر تعريف المستند من هذا النوع هو:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

التحقق من صحة صفحات XHTML

ننهي الحديث عن XHTML بالتذكير أنه بإمكان مطوِّر الويب (و غيره) أن يتحقق من مطابقة صفحة ويب ما لمعايير XHTML عبر الرابط:

http://www.w3schools.com/xhtml/default.asp

ملحوظة: يعكس وضع رابط التحقق من مطابقة الصفحة لمعايير XHTML في صفحتك نوعاً من الثقة بقدراتك البرمجية باعتقادي.

HTML5

ظهرت لغة HTML التي شرحناها في الصفحات السابقة (المعروفة في الأوساط التقنية بالإصدار HTML 4.01) في عام ١٩٩٩م، و في الحقيقة فإن تطورات كثيرة حصلت على مفهوم الويب منذ ذلك الحين إلى يومنا هذا، و أصبحنا نسمع بما يعرف بالـ Web 2.0، كما أصبح من النادر أن نجد موقع ويب (أعني المواقع الجيدة) لا يقدم خدمات مثل تشغيل الفيديو و الصوت و غير ذلك، و هذا ما استوجب تطوير إصدار جديد من لغة HTML بإضافة مجموعة جديدة من الوسوم و الواصفات إلى اللغة القديمة ليظهر الإصدار الجديد بالاسم HTML 5.

و من الميزات الجديدة التي جاءت بمجيء HTML 5 ظهور وسوم خاصة لعرض الفيديو و الصوت و وسوم خاصة لكتابة المقالات أو الأخبار و وسوم خاصة للرسم إضافةً إلى إضافة مجموعة جديدة من عناصر الإدخال الخاصة بالنماذج و غير ذلك مما سيعرض في الفقرات القادمة.

مستعرضات الويب الداعمة لـ HTML 5

لأن HTML 5 ظهرت منذ فترة قصيرة نسبياً (بدأ العمل على وضع معاييرها عام ٢٠٠٦) فإن معظم مستعرضات الويب القديمة لا تدعمها للأسف، و لكن المستعرضات الجديدة ستدعمها بكل تأكيد، فـ HTML 5 –برأيي– ستكون لغة بناء صفحات الويب المعيارية في المستقبل القريب و لهذا فقد بدأت بعض المستعرضات بدعمها فعلاً مثل الإصدارات الأخيرة من مستعرضات: Google Chrome و Firefox و Opera و Internet Explorer و Safari.[٦]

[٦] لا تعتقد أن دعم هذه المستعرضات كافٍ لجعل HTML 5 اللغة الرسمية لتطوير الصفحات فهناك عشرات مستعرضات الويب الأخرى و التي تستخدمها نسبة مرتفعة من مستخدمي الانترنت الإجماليين.

عنصر الفيديو Video

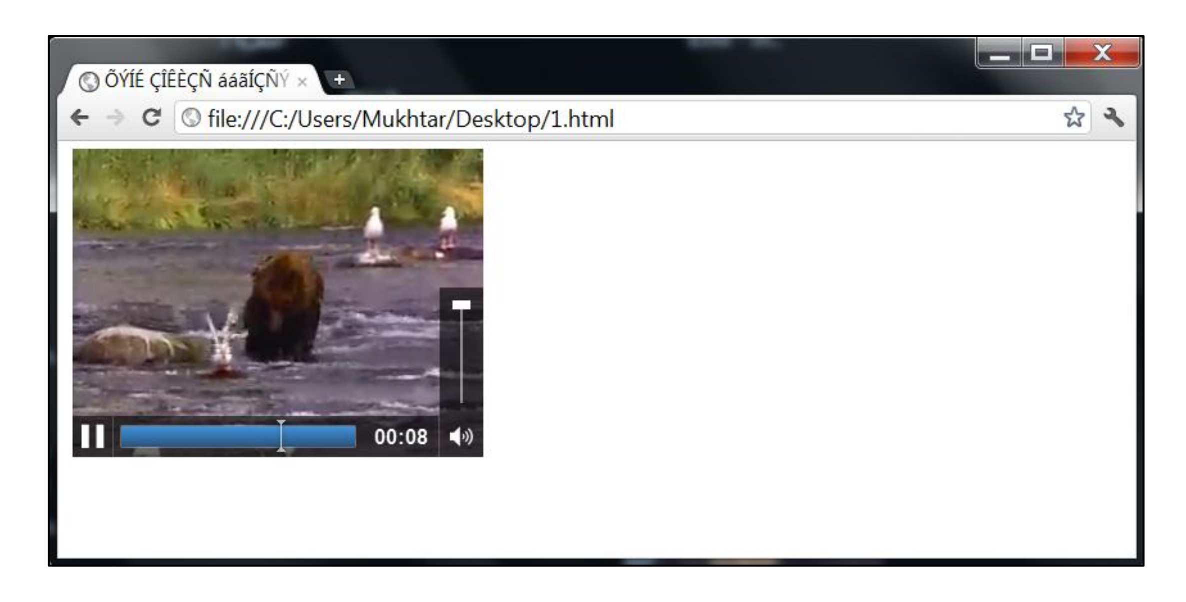
تقدم HTML 5 آلية بسيطة لعرض ملفات الفيديو ضمن صفحة الويب باستخدام العنصر الجديد <video>، و يتم ذلك بإسناد القيم المناسبة إلى واصفتي الارتفاع height و العرض width الخاصة به و من ثمَّ تمرير مسار ملف الفيديو عبر العنصر الابن <source> كقيمة لواصفته src و تمرير نوع ملف الفيديو عبر الواصفة type، لنشاهد الشيفرة التالية على سبيل المثال:

```
<html>
  <head>
        <title>
             مثال على عنصر عرض الفيديو
        </title>
  </head>
  <body>

    <video width="320" height="240" controls="controls">
        <source src="movie.mp4" type="video/mp4" />
        للأسف، المستعرض الحالي لا يدعم عنصر عرض الفيديو!
    </video>

  </body>
</html>
```

يظهر المحتوى النصي (الموجود بين وسمي البداية و النهاية للعنصر <video>) في المستعرضات التي لا تدعم HTML 5، أمَّا في المستعرضات الداعمة لها فستبدو صفحتنا كمايلي:



الشكل ٣١:   عنصر عرض الفيديو الجديد كما يبدو في مستعرض جوجل كروم

ملحوظة: إلى هذه اللحظة فإن العنصر <video> يدعم تشغيل ثلاثة أنواع من صيغ الفيديو و هي الصيغ MPEG4 و Ogg و WebM.

يعرض الجدول التالي واصفات العنصر <video>:

الواصفة	القيم الممكنة	الشرح
audio	muted	كتم الصوت
autoplay	autoplay	التشغيل التلقائي لملف الفيديو
controls	controls	عرض شريط الأدوات في مشغل الفيديو
height	قيمة بالبكسل	ارتفاع مشغل الفيديو
loop	loop	إعادة تشغيل الفيديو بشكل تلقائي عند انتهائه
poster	رابط تشعبي	رابط تشعبي لصورة تعبر عن مقطع الفيديو و تظهر قبل أن يقوم المستخدم بتشغيله
src	رابط تشعبي	رابط ملف الفيديو
width	قيمة بالبكسل	عرض مشغل الفيديو

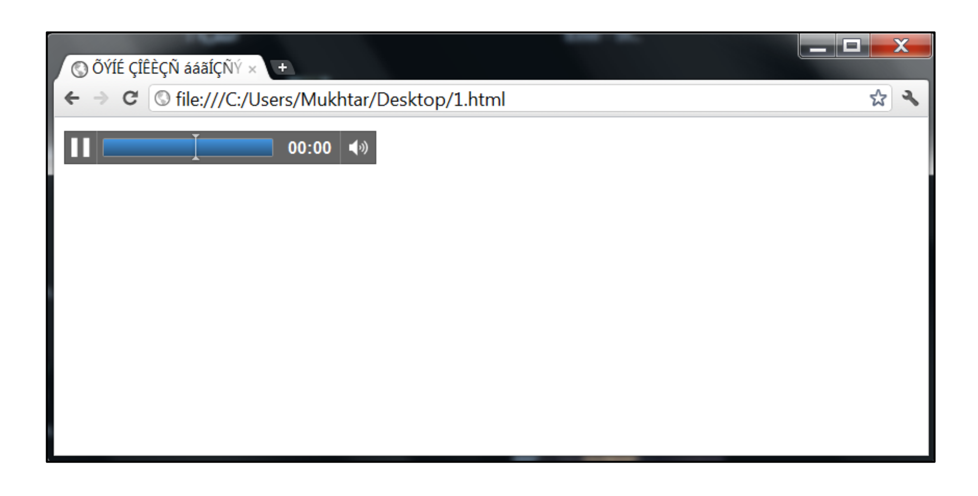
الجدول ٥ : واصفات العنصر <video>

عنصر الصوت Audio

تتيح HTML 5 عرض ملفات الصوت ضمن صفحة الويب ببساطة شديدة من خلال العنصر <audio> و الذي يستخدم بطريقة مشابهة لعنصر عرض الفيديو كمايلي:

```
<html>
  <head>
      <title>مثال على عنصر عرض الصوت</title>
  </head>
  <body>
     <audio controls="controls">
        <source src="song.mp3" type="audio/mp3" />
        للأسف، المستعرض الحالي لا يدعم عنصر عرض الصوت!
     </audio>
  </body>
</html>
```

يبدو هذا المثال في المستعرضات التي تدعم هذا العنصر كمايلي:



الشكل ٣٢ : عنصر عرض الصوت الجديد كما يبدو في مستعرض جوجل كروم

أمَّا في المستعرضات التي لا تدعم هذا العنصر فسيظهر النص المذكور بين وسمي البداية و النهاية للعنصر <audio> و الذي يحوي رسالة الاعتذار.

ملحوظة: إلى هذه اللحظة فإن العنصر <audio> يدعم تشغيل ثلاث صيغ لملفات الصوت هي MP3 و Ogg و Wav.

يعرض الجدول التالي واصفات العنصر <audio>:

الواصفة	القيم الممكنة	الشرح
autoplay	autoplay	التشغيل التلقائي لملف الصوت
controls	controls	عرض شريط الأدوات في مشغل الصوت
loop	loop	إعادة تشغيل ملف الصوت بشكل تلقائي عند انتهائه
preload	preload	تحميل الملف الصوت عند تحميل الصفحة بغض النظر عن خاصية التشغيل التقائي و عن نقر المستخدم لزر التشغيل
src	رابط تشعبي	رابط ملف الصوت

الجدول ٦ : واصفات العنصر <video>

عناصر النماذج Forms

تقدم HTML 5 مجموعة جديدة من عناصر الإدخال الخاصة بالنماذج Forms بالإضافة إلى مجموعة جديدة من أنواع عنصر الإدخال التقليدي <input>، و سنبدأ حديثنا بالحديث عن الأنواع الجديدة للعنصر <input> و التي يعرضها الجدول التالي:

قيمة الواصفة type	الشرح
email	عنصر خاص لإدخال البريد الإلكتروني
url	عنصر خاص لإدخال الروابط التشعبية
number	عنصر خاص لإدخال الأرقام

range	عنصر خاص لإدخال قيمة محصورة بمجال معيَّن سلفاً، و يظهر هذا العنصر على شكل شريط تمرير
date	عنصر لإدخال التاريخ (يوم/شهر/سنة)
month	عنصر لإدخال التاريخ (شهر/سنة)
week	عنصر لإدخال التاريخ و يستخدم لتحديد أسبوع من السنة
time	عنصر لإدخال الوقت (ساعة/دقيقة)
datetime	عنصر لإدخال التاريخ و الوقت (الساعة/اليوم/الشهر/السنة)، بالتوقيت العالمي
datetime-local	عنصر لإدخال التاريخ و الوقت، بالتوقيت المحلي
search	عنصر لعرض مربع البحث و هو شبيه بالنوع text
color	عنصر لإدخال القيم اللونيَّة

الجدول ٧ : أنواع عناصر الإدخال الجديدة في HTML 5

لنشاهد المثال التالي:

```
<!DOCTYPE HTML>
<html>
      <body>
            <form action="" method="get">
                  Name:
                  <input type="text" name="stdnt_name" /><br />
                  E-Mail:
                  <input type="email" name="stdnt_email" /><br />
                  Website:
                  <input type="url" name="stdnt_url" /><br />
                  Points:
```

```
                <input type="range" name="points" min="1"
max="10" /><br />

                Date:

                <input type="date" name="user_date" /><br />

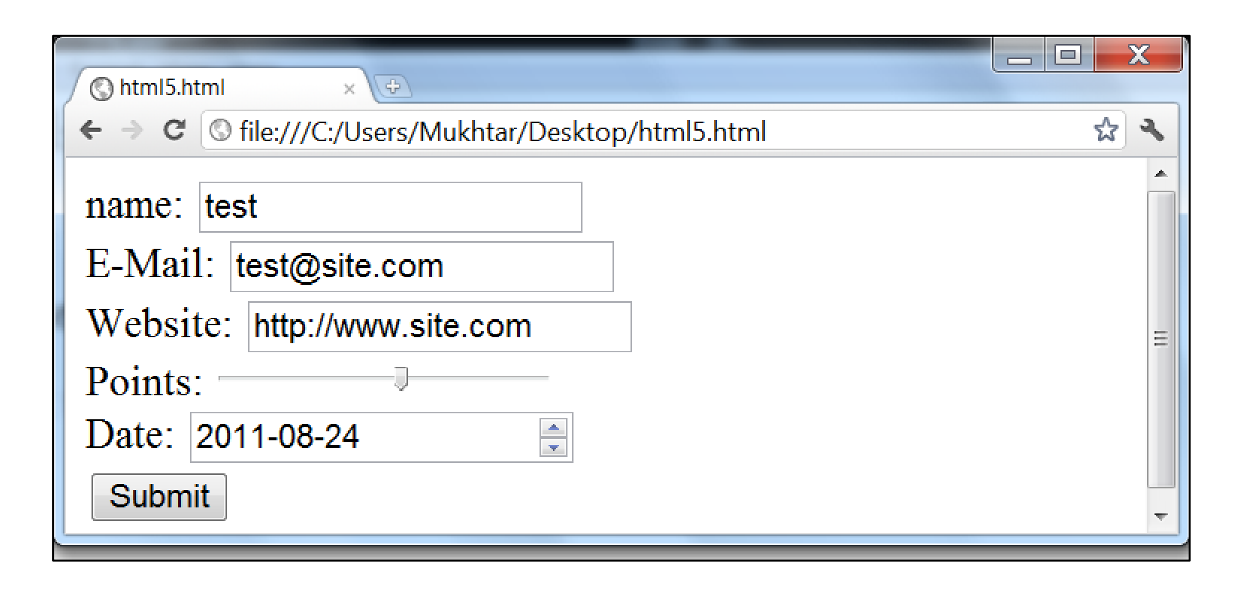
                <input type="submit" />

            </form>

        </body>

</html>
```

يبدو المثال السابق كمايلي في مستعرض الويب:



الشكل ٣٣: عناصر الإدخال الجديدة كما تبدو في مستعرض كروم

بالإضافة لما سبق ذكره تقدم HTML5 مجموعة من وسوم النماذج الجديدة و منها الوسم <datalist> و الذي يستخدم لإنشاء القوائم المنسدلة كخيارات إدخال افتراضيَّة لأحد عناصر الإدخال الموجودة و يُنشأ محتوى القائمة بالتعاون مع وسم <option> و بطريقة مشابهة للعنصر القديم <select>، كما يتم تمرير اسم قائمة الخيارات الافتراضية (عنصر <datalist>) لعنصر الإدخال عبر الواصفة الجديدة list، لنشاهد المثال التالي:

```
<!DOCTYPE HTML>

<html>

      <body>

            <form action="" method="get">

                  name:

                  <input type="text" list="names_list"
name="stdnt_name" /><br />

                  <datalist id="names_list">

                        <option label="A" value="nameA" />

                        <option label="B" value="nameB" />

                        <option label="C" value="nameC" />

                  </datalist>

            </form>

      </body>

</html>
```

من العناصر الجديدة التي قدمتها HTML5 أيضاً العنصران <keygen> و <output>، حيث يستخدم العنصر <keygen> في عمليات تسجيل الدخول و التحقق من المستخدمين Authentication و ذلك عبر توليد زوج مفاتيح (مفتاح عمومي/مفتاح خصوصي) يرسل مع الطلب دون تدخل المستخدم، و يستخدم بالشكل:

```
<keygen name="security" />
```

أما العنصر <output> فيستخدم لعرض مخرجات Outputs من أنواع مختلفة مثل نتائج الحسابات أو القيم المُعادَة من أحد توابع Java Script كمايلي:

```
<output id="result" onforminput="resCalc()"></output>
```

عنصر المَرسَم Canvas

تقدم HTML 5 عنصر المرسم <Canvas> و الذي يتيح لك الرسم ضمن صفحة الويب باستخدام Java Script [٧] و بذلك تستغني عن أية إضافات خارجية Plugins كإضافات الـ Flash على سبيل المثال.

لاستخدام عنصر المرسم يتم إنشاؤه أولاً بارتفاع height و عرض width محددين كمايلي:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

ثم يتم الرسم عليه باستخدام Java Script بشكل مشابه لمايلي:

```
<script type="text/javascript">

   var c=document.getElementById("myCanvas");
   var cxt=c.getContext("2d");
   cxt.fillStyle="#FF0000";
   cxt.fillRect(0,0,150,75);

</script>
```

حيث قمنا أولاً بتحديد عنصر المرسم عن طريق الـ id الخاص به و ذلك في السطر:

```
var c=document.getElementById("myCanvas");
```

ثمَّ قمنا بإنشاء كائن Object من المرسم و هيأناه للرسم ثنائي الأبعاد 2D في السطر التالي:

```
var cxt=c.getContext("2d");
```

الآن و للقيام بالرسم سيتم استدعاء طرق Methods الكائن cxt الذي تم إنشاؤه قبل قليل بالشكل:

```
cxt.fillStyle="#FF0000";
cxt.fillRect(0,0,150,75);
```

---

[٧] يمكنك قراءة المزيد عن Java Script في مرجع آخر فهي خارج نطاق هذا الكتاب و لكنني سأحاول شرح استخدامها في الأمثلة.

ففي السطر الأول تم تحديد اللون الأحمر لوناً للشكل الذي سيتم رسمه (هذا ما تعنيه القيمة #FF0000)[٨]، ثمَّ تم رسم المربع بتمرير إحداثيات ركنيه الأيسر العلوي و الأيمن السفلي بشكل مشابه لما فعلناه عند تحديد مناطق الخريطة الصورية (انظر فقرة الخرائط الصورية).

يبدو المثال السابق ضمن الشيفرة الكاملة لصفحة الويب كمايلي:

```
<!DOCTYPE HTML>
<html>
      <body>

         <canvas id="myCanvas" width="200" height="100">
                لا يدعم مستعرضك عنصر المرسم
         </canvas>

          <script type="text/javascript">
              var c=document.getElementById("myCanvas");
              var cxt=c.getContext("2d");
              cxt.fillStyle="#FF0000";
              cxt.fillRect(0,0,150,75);
          </script>
      </body>
</html>
```

يعرض الجدول التالي بعض طرق Methods كائن المرسم و وظيفة كل منها:

[٨] سيتم في الفصل الثاني عرض مفهوم الألوان في HTML.

الطريقة	الوظيفة	مثال
fillStyle	تحديد لون التعبئة	cxt.fillStyle="#FF0000";
fillRect	رسم مستطيل	cxt.fillRect(0,0,150,75);
moveTo	إزاحة مبدأ الإحداثيات	cxt.moveTo(10,10);
createLinearGradient	إنشاء تدرج لوني	cxt. createLinearGradient(0,0,170,50)
addColorStop	إضافة لون إلى التدرج اللون	cxt .addColorStop(0,"#FF0000");
drawImage	لرسم صورة	cxt.drawImage(img,0,0);
lineTo	لرسم خط مستقيم	cxt.lineTo(150,50);

الجدول ٨ : بعض طرق كائن العنصر <canvas>

تخزين البيانات من طرف العميل

تتيح HTML 5 كائنين مختلفتين لتخزين البيانات من طرف العميل Client و هما:

١. الكائن localStorage: يستخدم هذا الكائن لتخزين البيانات من طرف العميل لفترة غير محدودة و يستخدم الصيغة العامة التالية للتخزين:

```
localStorage.varName="value";
```

حيث أن varName هو اسم المتغير و value هي قيمة هذا المتغير و يمكن استرجاع قيمة المتغير بالصيغة العامة التالية:

```
localStorage.varName;
```

٢. الكائن sessionStorage: يستخدم هذا الكائن لتخزين البيانات من طرف العميل لفترة محددة بجلسة واحدة فقط، و يستخدم الصيغة العامة التالية للتخزين:

```
sessionStorage.varName="value";
```

حيث أن varName هو اسم المتغير و value هي قيمة هذا المتغير و يمكن استرجاع قيمة المتغير بالصيغة العامة التالية:

```
sessionStorage.varName;
```

ملحوظة: هذان الكائنان يستخدمان كمخازن خاصة للبيانات أثناء كتابة شيفرات Java Script.

دليل الوسوم الجديدة في HTML 5

ختاماً لهذا الفصل نعرض فيمايلي جدولاً بالوسوم الجديدة في HTML 5:

الوسم	الشرح
<article>	مقالة
<aside>	جانب من محتويات الموقع
<audio>	لتشغيل ملفات الصوت
<canvas>	للرسم برمجياً
<command>	زر أوامر
<datalist>	قائمة خيارات افتراضية لعنصر إدخال
<details>	تفاصيل عنصر ما
<embed>	تضمين عنصر تفاعلي أو plugin خارجي
<figcaption>	عنوان عنصر <figure>
<figure>	مجموعة من عناصر الوسائط
<footer>	ذيل منطقة من مناطق الصفحة
<header>	رأس منطقة من مناطق الصفحة
<hgroup>	معلومات عن قسم من المستند
<keygen>	توليد مفاتيح التحقق في النموذج
<mark>	نص مميز

<meter>	قياسات محصورة بمدى محدد مسبقاً
<nav>	روابط تشعبية
<output>	عرض خرج من نوع ما
<progress>	شريط تقدّم لمهمة من نوع ما
<rp>	يستخدم لغة Ruby للتصريح عن ما سيتم عرضه في حالة كون المستعرض لا يدعم عنصر Ruby
<rt>	شرح لما تم التصريح عنه باستخدام عنصر Ruby
<ruby>	عنصر Ruby
<section>	قسم من المستند
<source>	لتعريف مصادر الوسائط
<summary>	يعرف ترويسة العنصر detail
<time>	التاريخ/الوقت
<video>	لعرض عناصر الفيديو
<wbr>	سطر جديد محتمل

الجدول ٩ : دليل الوسوم الجديدة في HTML 5

الفصل الثانـي

CSS 3

مقدمة

تعلَّمنا في الفصل الماضي كيفيَّة كتابة مستندات الويب باستخدام لغة HTML5 و لكننا لم نتطرّق لنواحي كثيرة تتعلق بمظهر و تنسيق تلك المستندات، و في الحقيقة فقد كان هذا الأمر مقصوداً لأن ما يتعلق بالشكل و المظهر هو من مهمّة أوراق الأنماط الانسيابيَّة CSS التي يناقش هذا الفصل أحدث إصداراتها و الذي يعرف بـ CSS3، و لكننا –في واقع الأمر– لا نستطيع أن نتحدّث مباشرةً عن CSS3 لذلك سنتحدث قبل ذلك عن CSS التقليدية.

ما هي CSS ؟

إن CSS هي اختصار للأحرف الأولى من الجملة Cascading Style Sheet و التي تُرجِمَتْ على أنها "أوراق الأنماط الانسيابيَّة"[٩]، ففي حين أن HTML تهتم بوصف عناصر مستندات الويب فإن CSS تهتم بوصف شكل و مظهر هذه العناصر، و قد كان ظهور أوراق الأنماط الانسيابية الماحق الرئيسي لمشكلة من أكبر المشاكل التي واجهت مطوِّري الويب لسنوات قبل ظهورها، و الفقرة التالية تعرض هذه المشكلة.

مشكلة مطوري الويب قبل CSS

عندما تم تصميم لغة HTML كان التركيز الأكبر على تصميم لغة تصف عناصر صفحات الويب فقط، و لم يكن هناك أي توجه لدى مصممي HTML أن تهتم لغتهم بمظهر و شكل المحتوى المقدَّم، و تُرِكَتْ هذه المهمة لمصممي مستعرضات الويب، حيث كان –و ما يزال– كل مستعرض من مستعرضات الويب يعرض كل عنصر من عناصر صفحة الويب بمظهر افتراضي خاص به حسبما رأى مصممو المستعرض و بغض النظر عن رغبة مصمم الصفحة.

---

[٩] هذه الترجمة هي الغالبة على المراجع العربية.

تم التنبه لهذه النقطة لاحقاً فأضيفت وسوم جديدة تهتم بالتنسيق و من أمثلتها الوسم <font> الذي يستخدم لعرض نص بنوع خط (بنط)[١٠] Font و حجم و لون محدد ضمن الصفحة، و كان هذا شيئاً جميلاً في أول الأمر لكنه سرعان ما تحول لكارثة بالنسبة لمطوري الويب، فقد كان هناك بعض المواقع التي يصل عدد صفحاتها إلى مئات أو آلاف الصفحات أحياناً، مما يعني أن عملية تنسيق هذه الصفحات تحتاج لوقت و جهد كبيرين، ناهيك عن الحاجة إلى التركيز الشديد في ضبط خصائص الوسوم المذكورة سابقاً للحصول على صفحات بألوان و أحجام و أنواع خطوط موحدة و متجانسة، لكن و من حسن الحظ فقد تلاشت هذه المشكلة مع ظهور CSS.

مبدأيّاً يمكننا القول أن CSS تصف مظهر و شكل كل وسم من وسوم HTML على حدًى، و هذا ما يعني بالنسبة لمطور الويب أنه سيقوم بكتابة وصف لشكل و مظهر الوسم في مكان واحد و لمرة واحدة فقط، ثمَّ سيُطَبَّق هذا التنسيق على جميع مستندات الويب المرتبطة بذات الـ CSS.

البدء في استخدام CSS

هناك ثلاث طرق لاستخدام أوراق الأنماط الانسيابية CSS في صفحتك:

١. CSS السطريَّة Inline CSS:

يتم إسناد شيفرة CSS إلى الواصفة style الخاصة بكل عنصر على حدًى، انظر المثال التالي:

```
<p style="color:red;">محتوى الفقرة</p>
```

و هذه الطريقة غير مفضلة –بالنسبة لي– لأنها تعيدنا لمشاكل ما قبل CSS.

٢. CSS الداخليَّة (الدفينة) Embeded CSS:

---

[١٠] مع عدم اقتناعي بترجمة font إلى "بنط" لكنها وردت هكذا في معجم مصطلحات المعلوماتية الصادر عن الجمعية العلمية السورية للمعلوماتية و سأسمح لنفسي باستخدام "خط" كترجمة لـ font في كتابي هذا.

يتم كتابة شيفرة CSS في منطقة الـ Head من الصفحة بين وسمي <style> و </style> لتصبح الصفحة بالشكل:

```
<html>
      <head>
            <style type="text/css">
                  هنا يتم كتابة شيفرات css
            </style>
      </head>
      <body>
            محتوى الصفحة هنا
      </body>
</html>
```

و تعتبر هذه الطريقة جيدة للمواقع ذات العدد المحدود من الصفحات أما بالنسبة للمواقع المتوسطة و الكبيرة فأفضل استخدام الطريقة الثالثة.

٣. CSS الخارجيَّة External CSS:

يتم كتابة كامل شيفرة CSS في ملف مستقل يحمل الامتداد css. أولاً، ثمَّ يتم ربط هذا الملف مع صفحة الويب باستخدام الوسم <link> في منطقة رأس الصفحة عبر تمرير مسار الملف إلى واصفته href و تحديد نوع العلاقة بـ stylesheet كمايلي:

```
<html>
   <head>
      <link rel="stylesheet" type="text/css"
href="style.css" />
   </head>
```

```
   <body>محتوى الصفحة هنا</body>
</html>
```

و هذه الطريقة هي الطريقة المثالية لاستخدام CSS برأيي.

شيفرات CSS

تتكون شيفرة CSS من مجموعة من القواعد Rules، تصف كل قاعدة منها مظهر عنصر أو مجموعة من العناصر في الصفحة، و الشكل العام لقاعدة CSS هو:

```
selector {
            property:value;
            property:value;
            property:value;
            …
         }
```

و هذا ما يقوم بتطبيق جميع خصائص المظهر المذكورة بين القوسين } و { على كافة عناصر الصفحة التي تطابق المُحَدِّد selector.

ملحوظة: كما يبدو من اسمها فالمحددات تستخدم لتحديد عنصر أو مجموعة عناصر من صفحة الويب بغية تطبيق قواعد CSS عليها، و المُحَدِّدات كثيرة و متنوِّعة في CSS فقد يكون المُحَدِّد في أبسط حالاته اسماً لأحد وسوم HTML مثل h1 أو a أو p، و قد يكون المُحَدِّد id لعنصر ما مثل #link1 أو #header1 أو #searchForm و قد يكون غير ذلك، و بالطبع سيتم شرح مجموعة من المُحدِّدات بشكل ضمني في الفقرات التالية كما سيتم عرض جدول بكافة المحددات مع معانيها في نهاية الفصل.

يتم ذكر كل خاصية من خصائص CSS بشكل زوج اسم/قيمة بالشكل:

```
property:value;
```

حيث أن property هي اسم خاصية المظهر و value هي قيمة الخاصية، و ينتهي تعريف الخاصية بالفاصلة المنقوطة، و سنبدأ باستعراض كافة الخصائص اعتباراً من الفقرة التالية.

تذكِرة: كـ HTML فإن CSS لا تعطي أي قيمة لمحارف الفراغات و إنما يتم استخدامها لغرض تسهيل قراءة الشيفرة فقط.

خصائص مظهر النصوص

توفر CSS مجموعة من الخصائص لتنسيق مظهر النصوص و هي:

- خاصية المحاذاة text–align و التي تستخدم كمايلي:

```
h1 {text-align:center;}
```

حيث أن المحدد h1 يعني جميع عناصر <h1> في المستند، و يمكن إسناد القيمة right إلى الخاصية text-align لمحاذاة النص لليمين أو left لمحاذاة النص لليسار أو center لتوسيط النص أو justify لجعل جميع أسطر الفقرة متساوية العرض width.

- خاصية التزيين (التأثيث)[11] text–decoration و التي تستخدم كمايلي:

```
p a {text-decoration:overline;}
```

حيث أن المحدد p a يعني جميع عناصر الروابط <a> الموجودة ضمن أي من عناصر الفقرات <p> في المستند، و القيمة overline تعني عرض خط أعلى النصوص، و يمكن إسناد القيم التالية: none لإلغاء التزيين أو underline لعرض خط أسفل النص أو line–through لشطب النص.

- خاصية اتجاه القراءة direction و التي تستخدم كمايلي:

---

[11] هم يعربون Decor على أنها "أثاث" و لهذا أوردت التأثيث بين قوسين، علماً أنني أرى أن التزيين هي الترجمة الأنسب.

```
p {direction:rtl;}
```

لجعل اتجاه القراءة من اليمين إلى اليسار و هذا ما تعنيه القيمة rtl (اختصاراً لـ right to left)، و يمكن جعل اتجاه القراءة من اليسار لليمين بإسناد القيمة ltr إلى الواصفة direction.

- خاصية تحويل حالة الأحرف text–transformation و التي تستخدم كمايلي:

```
p {text-transformation:uppercase;}
```

لجعل جميع أحرف عناصر <p> بحالة الحرف الكبير[١٢]، و يمكن أن تأخذ هذه الخاصية القيم uppercase للحروف الكبيرة أو lowercase للحروف الصغيرة أو capitalize لجعل الحرف الأول فقط من كل كلمة حرفاً كبيراً.

- خاصية إزاحة البداية text–indent و التي تستخدم كمايلي:

```
p {text-indent:20px;}
```

لإزاحة أول كلمة من المقطع بمقدار ٢٠ بكسل، و الإزاحة غالباً ما تستخدم مع بداية فقرات اللغة الإنجليزيَّة.

- خاصية اللون color و التي تستخدم كمايلي:

```
p {color:red;}
```

حيث أن المحدد p يعني جميع العناصر <p> الموجودة في الصفحة، و يتم إسناد اللون للخاصية color إما بإسناد اسم اللون باللغة الإنجليزية مثل red أو green أو white ..إلخ، أو بإسناد قيمة اللون بالنظام الست عشري Hexadecimal مثل #FF0000 أو

---

[١٢] هناك لغات تعطي أكثر من رسم لنفس الحرف فمثلاً الحرف الأول في الإنجليزية يرسم A إذا كان في حالة الحرف الكبير و a في حالة الحرف الصغير.

‎#00FC00 أو #FFFFFF ..إلخ، أو بإسناد قيمة اللون عبر التابع rgb بالشكل rgb(255,0,0) أو rgb(0,255,0) أو rgb(255,255,255) ..إلخ.

توضيح: يتشكل كل لون من مزج ثلاثة ألوان هي الأحمر و الأخضر و الأزرق، و تختلف الألوان عن بعضها باختلاف نسبة كل من الأحمر/الأخضر/الأزرق فيها، و في حالة تمثيل اللون بالنظام الست عشري يتم اعتبار أول خانتين تمثلان درجة اللون الأحمر، و يتم اعتبار ثاني خانتين تمثلان درجة اللون الأخضر، و يتم اعتبار ثالث خانتين تمثلان درجة اللون الأزرق، و بالطبع تتدرج قيم كل خانة بين 00 كأدنى قيمة و FF كأعلى قيمة، فعلى سبيل المثال للحصول على اللون البنفسجي يجب مزج اللونين الأزرق و الأحمر دون أي نسبة من اللون الأخضر و بهذا تصبح قيمة اللون البنفسجي #FF00FF و يمكن التحكم بتدرج اللون عبر تخفيف درجة الأحمر و الأزرق لتصبحا على سبيل المثال #8B008B و هكذا، أما في حالة تمثيل اللون عبر التابع rgb فتمثل الخانة الأولى اللون الأحمر، و تمثل الثانية اللون الأخضر، و تمثل الثالثة اللون الأزرق لكن قيمة كل خانة تتدرج بين 0 كحد أدنى و 255 كحد أقصى.

لنشاهد المثال التالي و الذي يعرض استخدام خصائص مظهر النصوص:

```
<html>
  <head>
    <title>
       صفحة اختبار لخصائص مظهر النصوص
    </title>
    <style type="text/css">
       p {
            text-align:justify;
            color:#ff0000;
            text-decoration:underline;
            direction:rtl;
         }
```

```
    </style>

  </head>

  <body>

     <p>

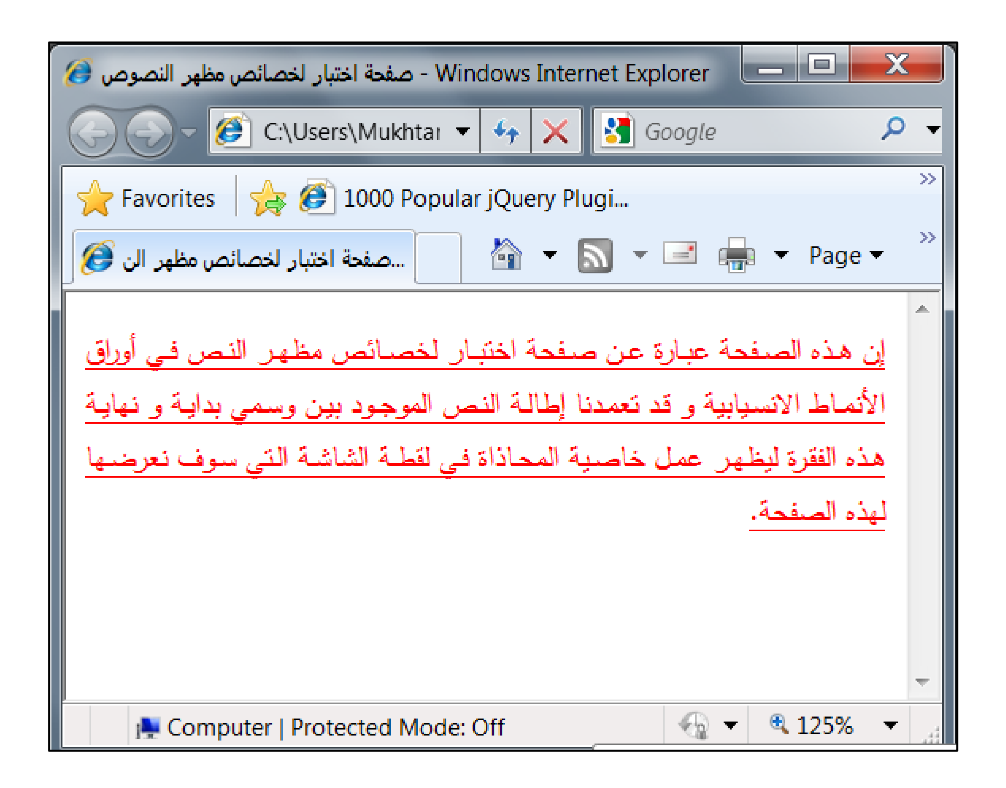
إن هذه الصفحة عبارة عن صفحة اختبار لخصائص مظهر النص في أوراق الأنماط الانسيابية و قد تعمدنا إطالة النص الموجود بين وسمي بداية و نهاية هذه الفقرة ليظهر عمل خاصية المحاذاة في لقطة الشاشة التي سوف نعرضها لهذه الصفحة.

     </p>

  </body>

</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



الشكل ٣٤ : استخدام خصائص مظهر النصوص

خصائص الخطوط Fonts

توفر CSS مجموعة من الخصائص للتحكم بالخطوط Fonts، و يتم التعامل مع الخطوط في CSS على شكل مجموعات تدعى كل منها بـ"عائلة خطوط" font-family، حيث يتم إسناد خط يتيم[١٣] أو أسماء أكثر من خط مفصولة بفواصل comma إلى الخاصية font-family و يكون الخط الأول هو الخط الافتراضي، و في الحالة التي يتعذر على المستعرض فيها العثور على الخط الافتراضي فإنه يجعل الخط التالي هو الافتراضي و هكذا.

توفر CSS ثلاثة عوائل افتراضية من الخطوط هي العائلة Serif و العائلة Sans-serif و العائلة Monospace و ينصح باستعمال العائلة الثانية للغة الإنجليزية و العائلة الثالثة عند عرض الشيفرات الحاسوبية مثل شيفرات لغات البرمجة في صفحة الويب.

كما توفر CSS مجموعة أخرى من الخصائص للتحكم بالخطوط هي الخاصية font-style و التي يمكن أن تسند إليها القيم italic (الخط المائل) و normal (الخط العادي)، و الخاصية font-size و التي تستخدم لتحديد حجم النص بالبكسل أو بالواحدة em (كل ١ em = ١٦ بكسل)، لنشاهد المثال التالي:

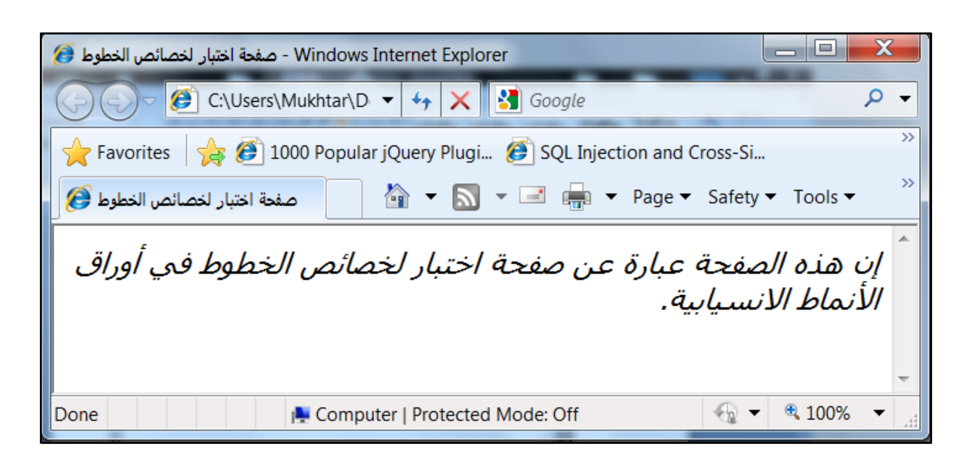
```
<html>
  <head>
    <title>
       صفحة اختبار لخصائص الخطوط
    </title>
    <style type="text/css">
```

[١٣] اجترحت مصطلح الخط اليتيم للحالة التي تكون فيها عائلة الخطوط مكونة من خط واحد.

```
        p {
             font-family:"Tahoma","Simplified Arabic",sans-serif;
             font-size:1.5em;
             font-style:italic;
             direction:rtl;
          }

    </style>
  </head>
  <body>
     <p>
إن هذه الصفحة عبارة عن صفحة اختبار لخصائص الخطوط في أوراق الأنماط الانسيابية.
     </p>
  </body>
</html>
```

و التي تبدو كمايلي في مستعرض الويب:



الشكل ٣٥ : استخدام خصائص الخطوط

ملحوظة: لم نضع علامتي اقتباس حول القيمة sans-serif لأننا نقصد عائلة الخطوط الافتراضية و التي تحمل اسم sans-serif.

خصائص مظهر الروابط

في الحقيقة لا توجد خصائص خاصة لمظهر الروابط، و لكن توجد محددات خاصة للروابط تستخدم لتنسيق مظهر الرابط في حالاته المختلفة، و المحددات هي:

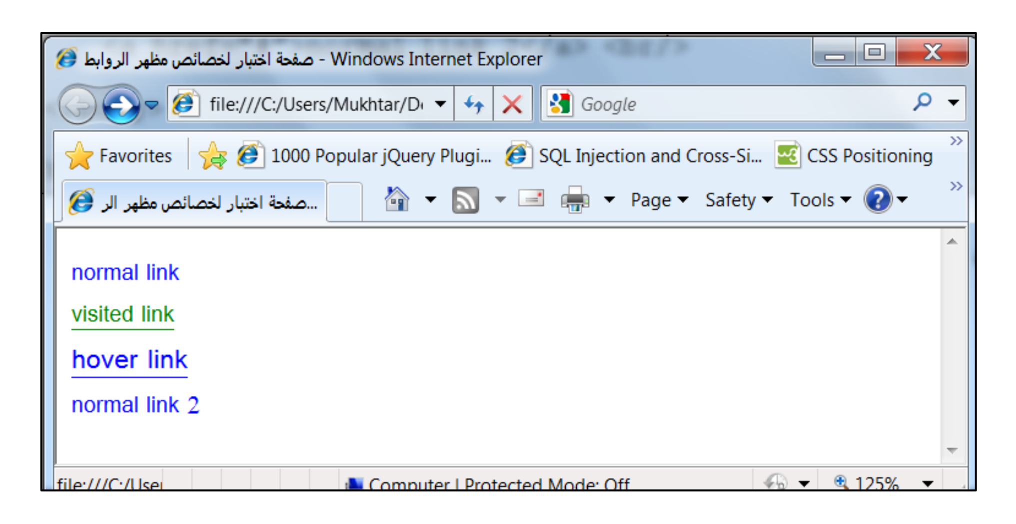
- a:link و الذي يستخدم لتنسيق مظهر الرابط في حالته العادية و هو مكافئ للمحدد a إذا استُخدِمَ وحده.
- a:visited و الذي يستخدم لتنسيق مظهر الرابط الذي تم النقر عليه مسبقاً.
- a:hove و الذي يستخدم لتنسيق مظهر الرابط عندما مرور مؤشّر الفأرة عليه.
- a:active و الذي يستخدم لتنسيق مظهر الرابط لحظة النقر عليه.

و المثال التالي يعرض استخدام هذه المحددات الخاصة:

```
<html>
  <head>
    <title>
        صفحة اختبار لخصائص مظهر الروابط
    </title>
    <style type="text/css">
        a:link{text-decoration:none;color:blue;}
        a:visited{color:green;}
        a:hover{text-decoration:underline;}
        a:active{font-size:120%;}
    </style>
```

```
  </head>
  <body>
     <a href="#">normal link</a> <br/>
     <a href="www.google.com">visited link</a> <br/>
     <a href="#">hover link</a> <br/>
     <a href="#">normal link 2</a> <br/>
  </body>
</html>
```

يبدو المثال السابق في صفحة الويب بعد زيارة بعض الروابط:



الشكل ٣٦ : هكذا تبدو صفحة اختبار مظهر الروابط لحظة النقر على الرابط الثالث و بعد زيارة الثاني

ملحوظة: يجب ذكر قاعدة a:hover بعد a:link و a:visited، كما يجب ذكر a:active بعد a:hover.

خصائص الخلفيات Backgrounds

توفر CSS مجموعة من الخصائص لتنسيق خلفيات العناصر، نبدأ استعراضها بالحديث عن الخاصية background-color و التي تستخدم لتحديد خلفية لونيَّة لأحد العناصر بإسناد قيمة اللون إليها بالشكل:

```
background-color:#CF1A11;
```

و يمكن تعيين صورة كخلفية لأحد العناصر عبر إسناد مسار الصورة للخاصية background–image بالشكل:

```
background-image:url('image.jpg');
```

و في حال كون حجم الصورة (الخلفيَّة) أصغر من حجم العنصر يتم تحديد طريقة تكرار الصورة عبر الخاصة background–repeat فيتم إسناد القيمة repeat–x لتكرار الصورة بشكل أفقي أو القيمة repeat–y لتكرار الصورة بشكل عمودي أو repeat لتكرارها بشكل أفقي و عمودي معاً أو no–repeat لعدم التكرار و هذا في الحالة الأولى، أما في الحالة الثانية فيمكن عدم تكرار الصورة و تحديد موضعها عن طريق الخاصية background–position حيث يتم إسناد واحدة أو أكثر من القيم left top أو left center أو left bottom أو center top أو center center أو center bottom أو right top أو right center أو right bottom، لنشاهد المثال التالي و الذي سنعرض فيه عنصرين <p> الأول بخلفية لونية و الآخر بخلفية صورية:

```
<html>
  <head>
    <title>
        صفحة اختبار لخصائص الخلفية
    </title>
    <style type="text/css">
        .firstP {
                    background-color:black;
                    color:white;
                }
        .secondP {
```

```
                    background-image:url('image1.jpg');
                    background-repeat:no-repeat;
                     background-position:center center;
                 }
        p {direction:rtl;}
    </style>
  </head>
  <body>
      <p class="firstP">
           العنصر الأول سيبدو بخلفية سوداء و لون نص أبيض
      </p>
      <p class="secondP">
           العنصر الثاني سيبدو بخلفية صورية و لهذا <br/><br/>
           سأقوم بالنزول بضعة أسطر بغية منح هذا العنصر <br/>
           <br/>
           المساحة الكافية لعرض الصورة الخلفية
      </p>
  </body>
</html>
```

ملحوظة: هناك طريقة خاصة لتحديد العناصر تعرف باسم التحديد بالفئات classes حيث يتم استبدال المحدد بأي اسم مسبوقاً بنقطة و هذا ما يعرف بتعريف الفئة class في CSS، بعد ذلك يتم إسناد اسم هذه الفئة (دون نقطة) إلى الواصفة class للعناصر التي نرغب بجعل مظهرها مطابقاً للخصائص المذكورة في قاعدة الفئة.

يبدو المثال السابق في مستعرض الويب كمايلي:



الشكل ٣٧ : هكذا تبدو صفحة اختبار خصائص الخلفية

خصائص مظهر القوائم

يتم تخصيص مظهر القوائم في CSS عبر تخصيص نوع الرمز/الرقم المعروض أمام كل عنصر من عناصر القائمة، و يتم هذا من خلال الخاصيتين التاليتين:

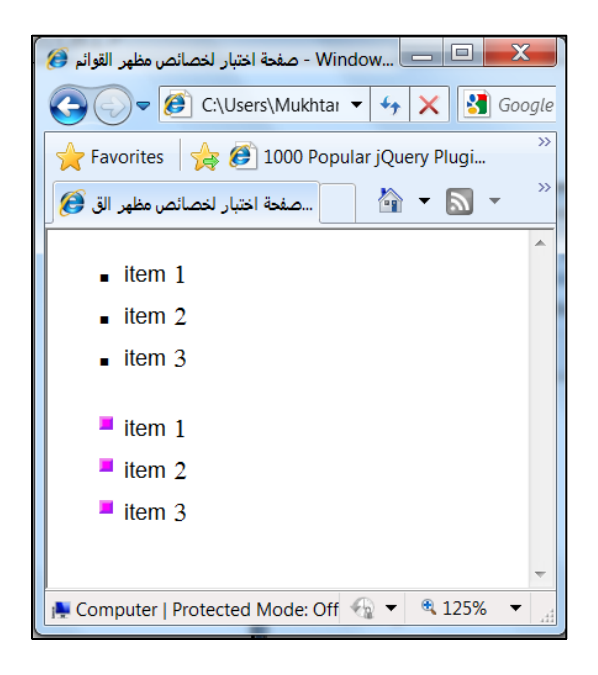
- list-style-type: و التي تستخدم لتحديد نوع الرمز/الترقيم أمام كل عنصر من عناصر القائمة، و تأخذ القيم circle و square و disc و none في حالة القائمة غير المرتبة، و القيم upper-roman و lower-roman و lower-alpha و upper-alpha و lower-latin و upper-latin و decimal و غيرها في حالة القائمة المرتبة.
- list-style-image: و التي تستخدم لوضع صورة خاصة أمام كل عنصر من عناصر القائمة كرمز له، و تستخدم بالشكل:

```
list-style-image:url('image.gif');
```

لنشاهد المثال التالي:

```
<html>
  <head>
    <title>
        صفحة اختبار لخصائص مظهر القوائم
    </title>
    <style type="text/css">
        ul {list-style-type:square;}
        ol {list-style-image:url('dot.gif');}
    </style>
  </head>
  <body>
     <ul>
          <li>item 1</li>
          <li>item 2</li>
          <li>item 3</li>
     </ul>
     <ol>
          <li>item 1</li>
          <li>item 2</li>
          <li>item 3</li>
     </ol>
  </body>
</html>
```

يبدو هذا المثال في مستعرض الويب كمايلي:



الشكل ٣٨ : هكذا تبدو صفحة اختبار خصائص مظهر القوائم في مستعرض الويب

خصائص مظهر الجداول

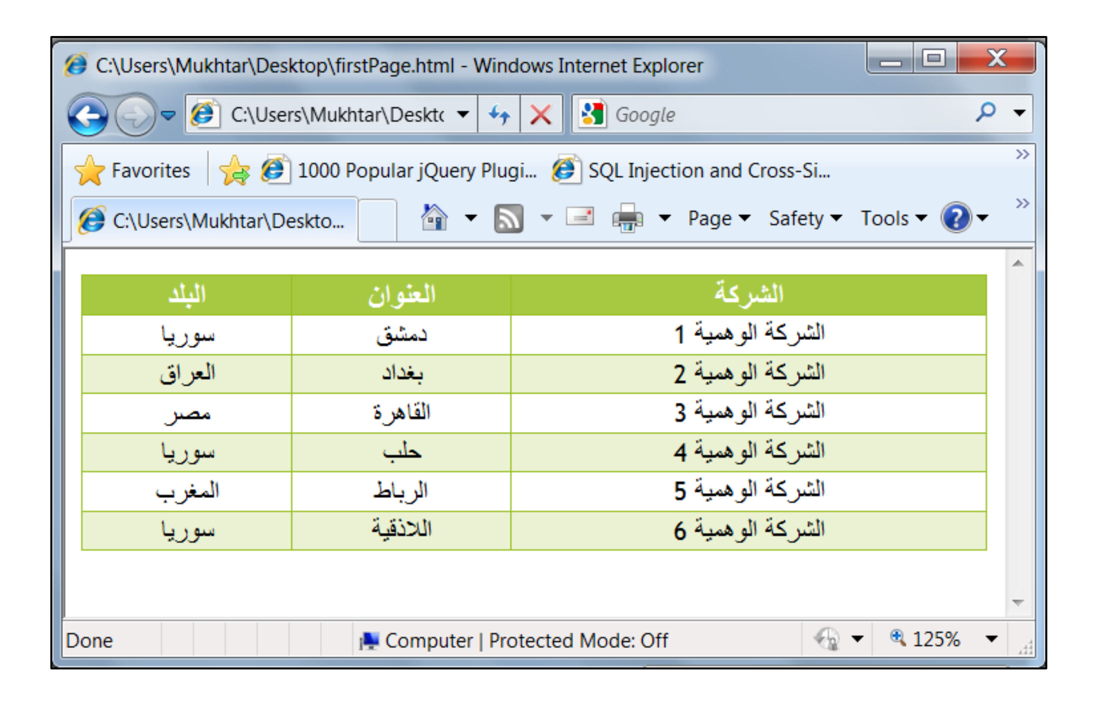
أظن أن الجداول هي أكثر عناصر HTML التي تستفيد من CSS إذا عرف مطور الويب كيفية توظيف خصائص CSS التي عُرِضَتْ سابقاً بالشكل المناسب، لنشاهد المثال التالي أولاً:

```
<html>
  <head>
    <style type="text/css">
      table
      {
            font-family:"Trebuchet MS", sans-serif;
            width:100%;
            border-collapse:collapse;
```

```
          direction:rtl;
          text-align:center;
     }
     td, th
     {
          font-size:1em;
          border:1px solid #98bf21;
     }
     th
     {
          font-size:1.1em;
          background-color:#A7C942;
          color:#ffffff;
     }
     .alt td
     {
          color:#000000;
          background-color:#EAF2D3;
     }
   </style>
</head>
<body>
          <table>
               <tr>
                  <th>الشركة</th>
```

```
                    <th>العنوان</th>
                    <th>البلد</th>
                </tr>
                <tr>
                    <td>الشركة الوهمية ١</td>
                    <td>دمشق</td>
                    <td>سوريا</td>
                </tr>
                <tr class="alt">
                    <td>الشركة الوهمية ٢</td>
                    <td>بغداد</td>
                    <td>العراق</td>
                </tr>
                <tr>
                 …
                 إلى آخر شيفرة الجدول
                 …
            </table>
  </body>
</html>
```

يبدو هذا المثال في مستعرض الويب كمايلي:



البلد	العنوان	الشركة
سوريا	دمشق	الشركة الوهمية 1
العراق	بغداد	الشركة الوهمية 2
مصر	القاهرة	الشركة الوهمية 3
سوريا	حلب	الشركة الوهمية 4
المغرب	الرباط	الشركة الوهمية 5
سوريا	اللاذقية	الشركة الوهمية 6

الشكل ٣٩ : توظيف CSS لتنسيق مظهر الجداول

و ما يجب ذكره من خصائص CSS عند الحديث عن المثال السابق هو مايلي:

- border: و التي يسند إليها قيمة بالبكسل تعبر عن سماكة حدود الجدول و لون يعبر عن لون الحدود بالإضافة إلى تحديد نوع الحدود (خط متصل solid، خط متقطع dashed، منقط dotted .. إلخ)، و بالطبع يفصل محرف المسافة بين هذه القيم (سيتم الحديث عن هذه الخاصية بشكل أكثر تفصيلاً في فقرة لاحقة).
- border–collapse: و التي تحدد هل سيتم طي حدود الخلايا و اعتبار وجود حد واحد بين كل خليتين (القيمة collapse) أم سيتم اعتبار حد كل خلية مستقلاً عن الآخر.

ملحوظة: لتسهيل قراءة محتويات الجدول ينصح دائماً بجعل أسطر الجدول متناوبة ألوان الخلفية و ألوان النصوص كما فعلنا في الجدول المعروض في مثالنا السابق، و هذا ما حققناه عبر استخدام الفئة alt. ، يعرف هذا التناوب اللوني باسم Zebra-stripping

خصائص الموضَعة Positioning

تتيح خصائص الموضَعة في CSS تحديد موضِع العنصر، و كيف يتم عرض العنصر عندما يكون محتواه كبيراً جداً، كما يتيح وضع عناصر خلف عناصر أخرى، و هناك عدة أنواع لتحديد موضع العنصر في CSS في الحقيقة و هي:

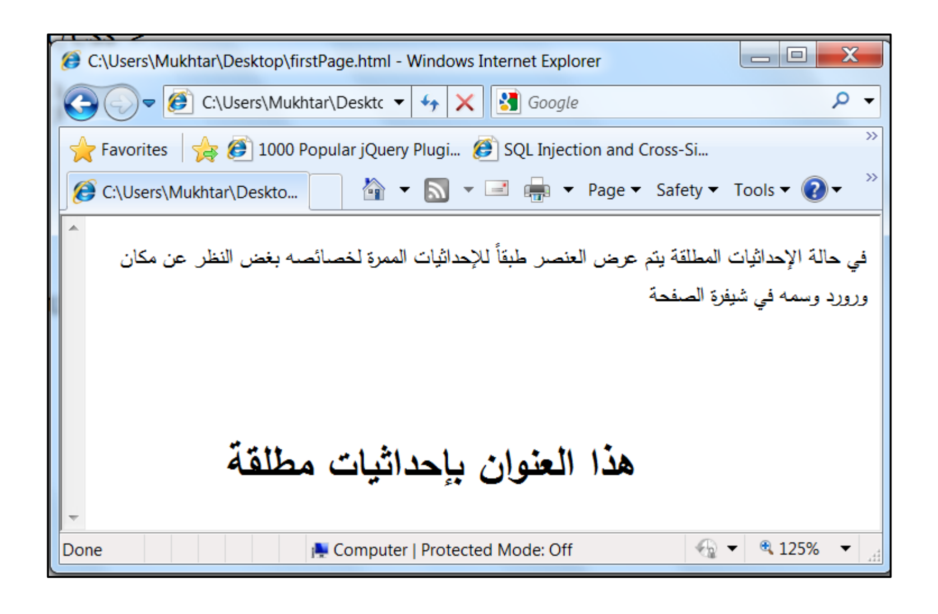
- النوع الستاتيكي Static: يتم وضع العنصر في مكان ظهوره في الصفحة أي بحسب ترتيب و موضع ظهور وسمه ضمن وسوم الصفحة، و هذا النوع هو الافتراضي.
- النوع الثابت Fixed: يتم وضع العنصر طبقاً لإحداثيات ثابتة يتم إسنادها بالبكسل للخصائص top و left و right و bottom مع إسناد القيمة fixed إلى الخاصية position، و بالطبع يتم اعتبار الركن العلوي الأيسر للصفحة مركزاً للإحداثيات.
- النوع النسبي Relative: يتم وضع العنصر طبقاً لإحداثيات ثابتة يتم إسنادها بالبسكل للخصائص top و left و right و bottom مع إسناد القيمة relative إلى الخاصية position، و يتم اعتبار الركن العلوي الأيسر للعنصر نفسه في حالته الستاتيكية مركزاً للإحداثيات.
- النوع المُطْلَق Absolute: يتم وضع العنصر طبقاً لإحداثيات ثابتة يتم إسنادها بالبكسل للخصائص top و left و right و bottom مع إسناد القيمة absolute إلى الخاصية position، و يتم اعتبار الركن الأيسر العلوي لأول عنصر أب بموضع غير ستاتيكي مركزاً للإحداثيات، و في حال عم وجود أي عنصر أب بموضع غير ستاتيكي يتم اعتبار الركن الأيسر العلوي للعنصر <html> مركزاً للإحداثيات.

لنشاهد المثال التالي:

```
<html>
  <head>
```

```
    <style type="text/css">
        * {direction:rtl;}
        h1
        { position:absolute; left:100px; top:150px; }
    </style>
  </head>
  <body>
    <h1>هذا العنوان بإحداثيات مطلقة</h1>
    <p>في حالة الإحداثيات المطلقة يتم عرض العنصر طبقاً للإحداثيات الممرة لخصائصه بغض النظر عن مكان وورود وسمه في شيفرة الصفحة</p>
  </body>
</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



الشكل ٤٠ : مثال على استخدام خصائص موضَعَة العناصر في CSS

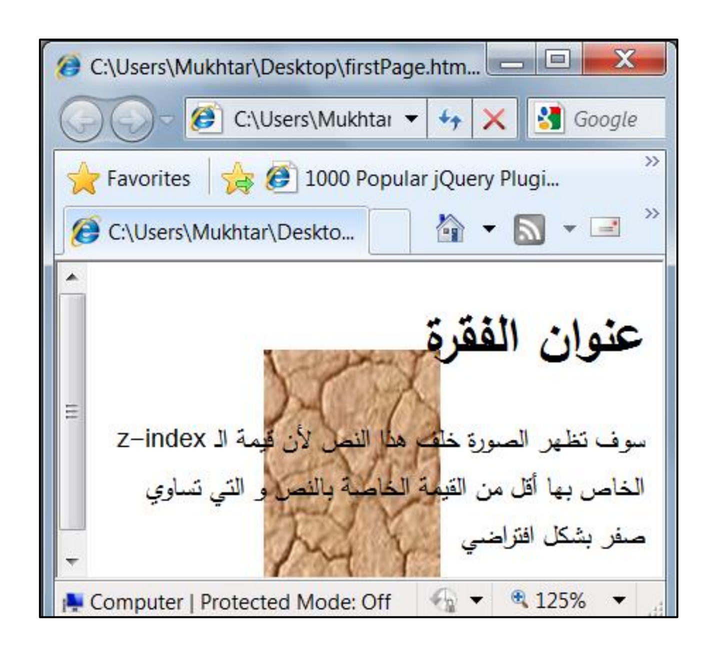
ملحوظة: المحدد الخاص * يقوم بتحديد جميع عناصر المستند.

يمكن وضع عناصر خلف عناصر أخرى عن طريق الخاصية z-index و ذلك بإسناد رقم صحيح إليها، حيث يتم إظهار العنصر ذو الـ z-index الأعلى أمام العناصر الأخرى، لنشاهد المثال التالي:

```
<html>
  <head>
    <style type="text/css">
      * {direction:rtl;}
      img
      {
        position:absolute;
        left:100px;
        top:50px;
        z-index:-1;
      }
    </style>
  </head>
  <body>
    <h1>عنوان الفقرة</h1>
    <img src="image1.jpg" width="100" height="140" />
    <p>سوف تظهر الصورة خلف هذا النص لأن قيمة الـــ
        z-index
      الخاص بها أقل من القيمة الخاصة بالنص و التي تساوي صفر
      بشكل افتراضي
    </p>
```

```
  </body>
</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



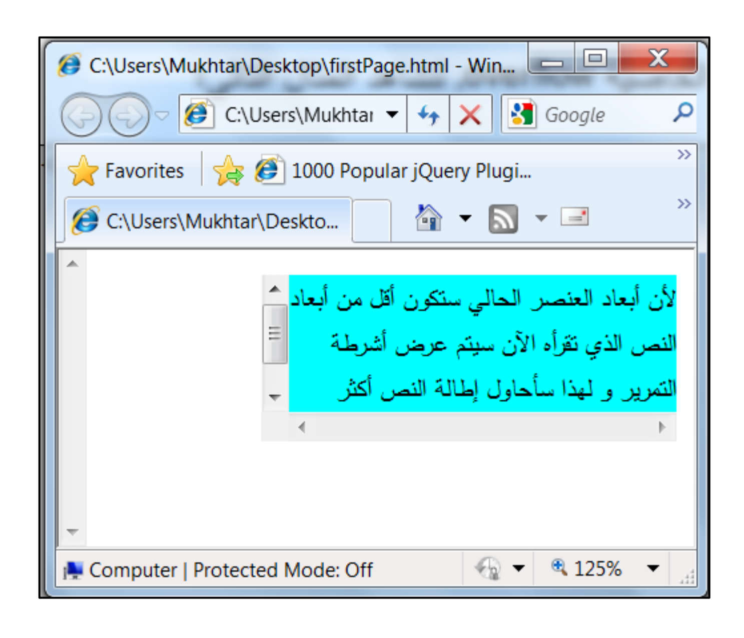
الشكل ٤١ : مثال على استخدام الخاصية z-index

يمكن أيضاً عرض أشرطة تمرير في الحالة التي يكون فيها محتوى العنصر أكبر بكثير من حجمه، و ذلك عن طريق إسناد القيمة scroll إلى الخاصية overflow، لنشاهد المثال التالي:

```
<html>
  <head>
    <style type="text/css">
      * {direction:rtl;}
      div
      {
         background-color:#00FFFF;
         width:250px;
```

```
        height:100px;
        overflow:scroll;
      }
    </style>
  </head>
  <body>
      <div>
          <p>لأن أبعاد العنصر الحالي ستكون أقل من أبعاد النص الذي تقرأه الآن سيتم عرض أشرطة التمرير و لهذا سأحاول إطالة النص أكثر بإضافة بعض الكلمات الإضافية إليه.</p>
      </div>
  </body>
</html>
```

يبدو المثال السابق كمايلي في مستعرض الويب:



الشكل ٤٢ : مثال على عرض أشرطة التمرير للعناصر ذات المحتوى الكبير

واحدات القياس في CSS

يتم تحديد أبعاد العناصر على سبيل المثال من خلال خاصيتي العرض width و الارتفاع height وذلك بإسناد قيم رقمية بالبكسل أو بإحدى واحدات القياس التالية:

الواحدة	الشرح	مثال
px	بكسل	150px
%	بالنسبة المئوية	50%
em	كل ١ em يساوي ١٦ بكسل	1.5em
in	إنش	1in
cm	سنتي ميتر	1cm
mm	ميلي ميتر	100mm
pt	نقطة (النقطة = $\frac{1}{72}$ إنش)	30pt
pc	بيكا (البيكا = ١٢ نقطة = $\frac{1}{6}$ إنش)	15pc

الجدول ١٠ : واحدات القياس في CSS

نموذج الصندوق CSS Box Model

عند الحديث عن تصميم صفحات الويب يجب في الحقيقة فهم ما يعرف بـ"نموذج الصندوق" أو box model بالإنجليزيَّة، حيث يتم تمثيل كل عنصر من عناصر مستند HTML عند إنشائه في المستعرض بأربعة مستطيلات متداخلة تشكل نموذج الصندوق و هي من الداخل إلى الخارج:

١- محتوى العنصر content: و يمثل كل ما هو موجود بين وسمي بداية و نهاية العنصر.

٢- الإزاحة الداخلية padding: و تمثل المسافة الفاصلة بين محتوى العنصر و بين حدوده.

٣- الحدود border: و يمثل الإطار الخارجي للعنصر.

٤- الإزاحة الخارجية margin: و تمثل المسافة الفاصلة بين حدود العنصر و العناصر الأخرى في المستند.

يوضِّح الشكل التالي نموذج الصندوق:



الشكل ٤٣ : نموذج الصندوق box model

تتيح CSS لمطوِّر الويب التحكم الكامل بكل منطقة من هذه المناطق الأربع عبر مجموعة من الخصائص نستعرضها في الفقرات التالية.

خصائص الحدود border

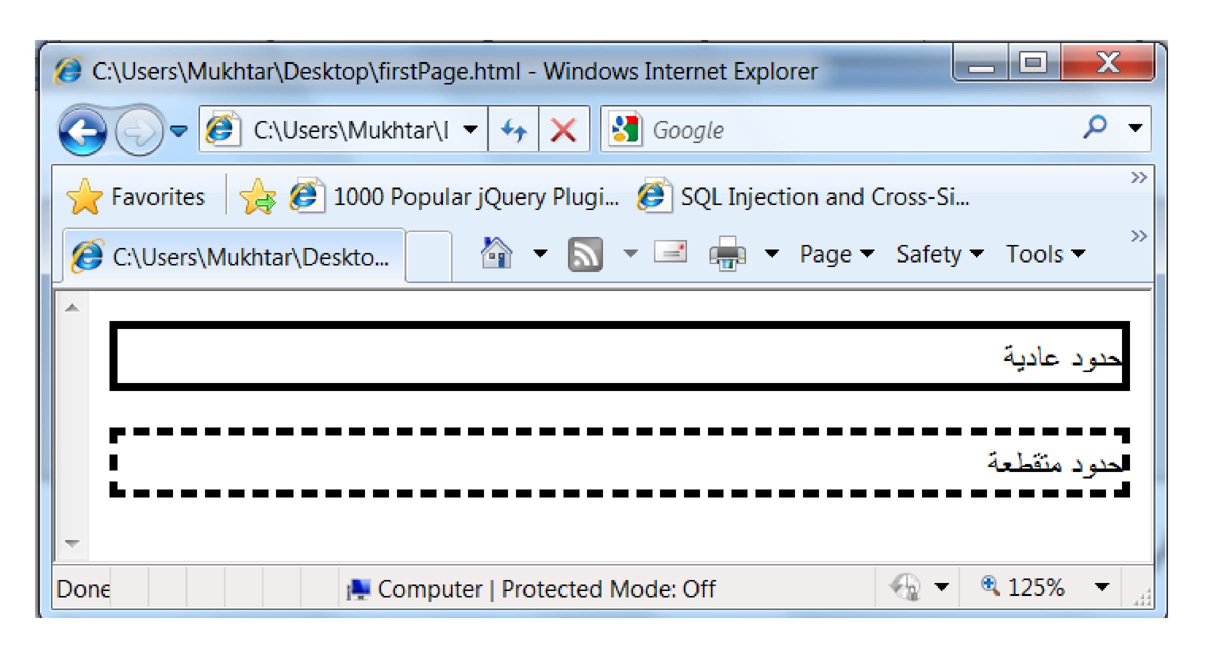
تسمح CSS لمطور الويب أن يتحكم بحدود العناصر عبر الخصائص border-width و border-style و border-color، يتم إسناد قيمة رقمية بإحدى واحدات القياس إلى الخاصة border-width تعبر عن مقدار سماكة الحدود، و يتم إسناد قيمة نصية تعبر عن شكل الحدود للخاصية border-style و التي يمكن أن تأخذ إحدى القيم dotted للحدود المنقطة و dashed للحدود المتقطعة و solid للحدود العادية و double للحدود المضاعفة و groove و ridge و inset و outset للحدود ثلاثية الأبعاد، و يتم إسناد قيمة لونية تعبر عن لون الحدود، لنشاهد المثال التالي:

```
<html>

  <head>
```

```
    <style type="text/css">
      * { direction:rtl; }
      .one
      { border-style:solid; }
      .two
      { border-style:dashed; }
    </style>
  </head>
  <body>
    <p class="one">حدود عادية</p>
    <p class="two">حدود متقطعة</p>
  </body>
</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



ملحوظة: يمكن تحديد شكل border–style خاص لكل طرف من أطراف الحدود و ذلك عن طريق إسناد أربعة قيم مفصولة بفراغات للخاصية border–style تمثل الأولى شكل الحد العلوي و الثانية شكل الحد الأيمن و الثالثة شكل الحد السفلي و الرابعة شكل الحد الأيسر بالشكل:

```
border–style: dotted dashed groove double;
```

أو عن طريق الخصائص الفردية border–left–style و border–right–style و border–top–style و border–bottom–style، و بالطبع فإن هذه الملحوظة تنطبق على خاصيتي اللون border–color و العرض border–width.

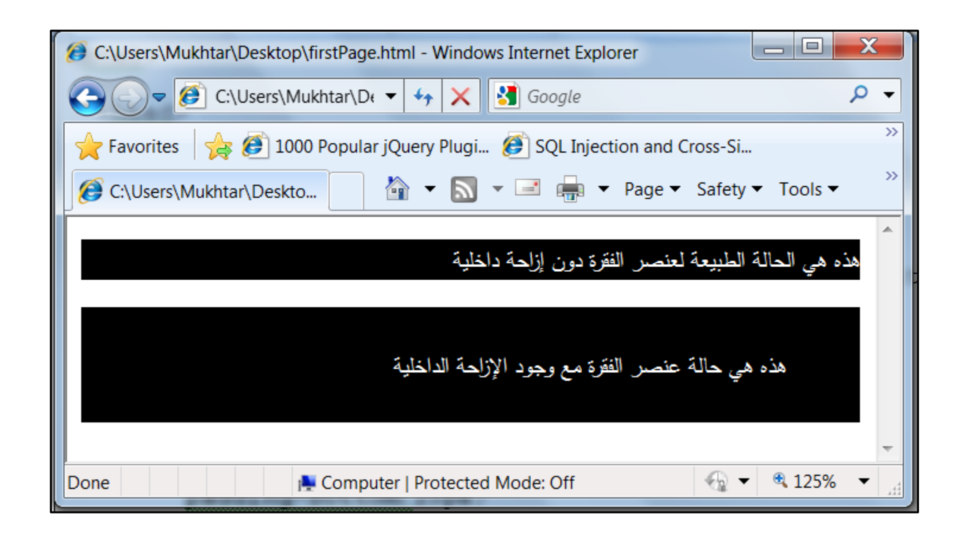
خصائص الإزاحة الداخلية padding

يتم تحديد مقدار الإزاحة الداخلية (المسافة بين محتوى العنصر و حدوده) عن طريق الخصائص padding–left لإزاحة الطرف الأيسر و padding–right لإزاحة الطرف الأيمن و padding–top لإزاحة الطرف العلوي و padding–bottom لإزاحة الطرف السفلي، أو عن طريق الخاصية padding لإزاحة جميع الأطراف، و يُسنَد إلى هذه الخصائص قيم بإحدى واحدات القياس تعبر عن مقدار الإزاحة الداخلية، لنشاهد المثال التالي:

```
<html>
  <head>
    <style type="text/css">
      p
      { direction:rtl; background-color:black; color:white; }
      .padding
      {
        padding-top:25px;
        padding-bottom:25px;
        padding-right:50px;
```

```
         padding-left:50px;
       }
    </style>
  </head>
  <body>
    <p> هذه هي الحالة الطبيعة لعنصر الفقرة دون إزاحة داخلية</p>
    <p class="padding">هذه هي حالة عنصر الفقرة مع وجود الإزاحة الداخلية</p>
  </body>
</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



الشكل ٤٤ : مثال على استخدام خصائص الإزاحة الداخلية

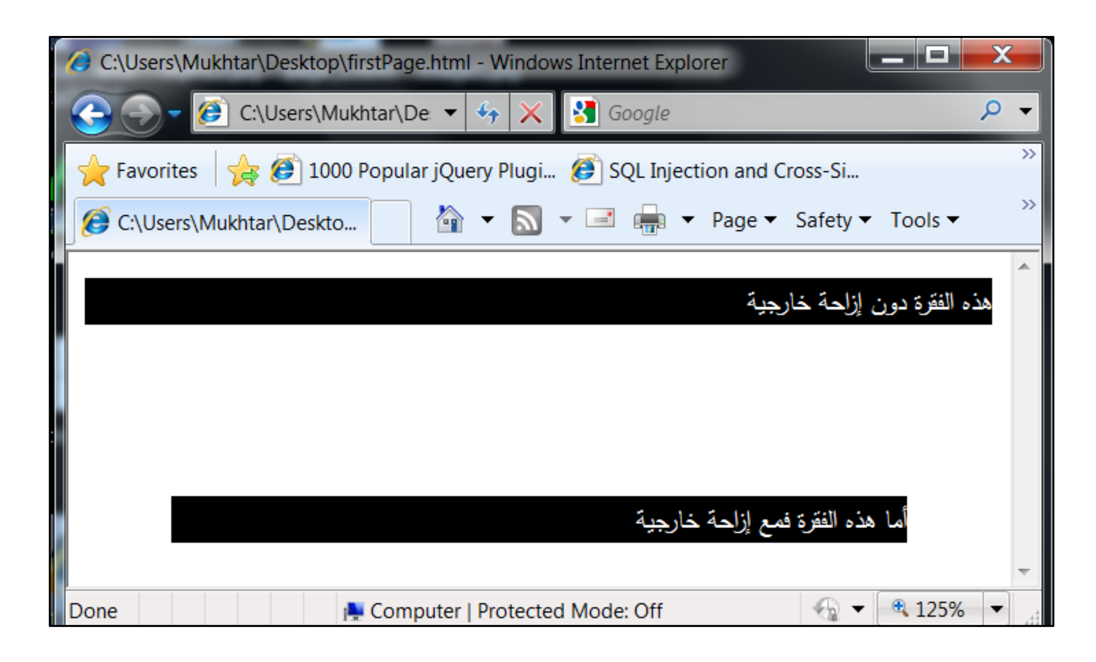
خصائص الإزاحة الخارجية margin

يتم تحديد مقدار الإزاحة الخارجية (المسافة بين حدود العنصر و العناصر الأخرى) عن طريق الخصائص margin-left لإزاحة الطرف الأيسر و margin-right لإزاحة الطرف الأيمن و

margin-top لإزاحة الطرف العلوي و margin-bottom لإزاحة الطرف السفلي، أو عن طريق الخاصية margin لإزاحة جميع الأطراف، و يُسنَد إلى هذه الخصائص قيم بإحدى واحدات القياس تعبر عن مقدار الإزاحة الداخلية أو القيمة الخاصة auto لترك هذه المهمة لمستعرض الويب، لنشاهد المثال التالي:

```
<html>
  <head>
    <style type="text/css">
      p
      { direction:rtl; background-color:black; color:white; }
      .margin
      {
        margin-top:100px;
        margin-bottom:100px;
        margin-right:50px;
        margin-left:50px;
      }
    </style>
  </head>
  <body>
    <p>هذه الفقرة دون إزاحة خارجية</p>
    <p class="margin">أما هذه الفقرة فمع إزاحة خارجية</p>
  </body>
</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



الشكل ٤٥: مثال على استخدام خاصية الإزاحة الخارجة **margin**

إخفاء العناصر

من الممكن إخفاء عناصر HTML باستخدام CSS بطريقتين:

- إسناد القيمة none إلى الخاصية display.
- إسناد القيمة hidden إلى الخاصية visibility.

و الاختلاف بينهما أن الطريقة الثانية تخفي العنصر مع الحفاظ على الحيز الذي يشغله فارغاً أما الطريقة الأولى تخفي العنصر مع عدم الحفاظ على الحيز الذي يشغله مما يؤثر على تموضع العناصر.

أنواع عرض العناصر

يتم عرض العناصر في HTML بطريقتين:

- بشكل كتلة block: حيث يتم النزول سطراً قبل العنصر و النزول سطراً بعد العنصر و هذه الحالة نراها مع عناصر مثل <h1> و <p> و <div>...إلخ.
- بشكل سطري inline: حيث يتم وضع العنصر ضمن نفس السطر دون الحاجة للنزول سطراً و هذه الحالة نراها مع عناصر مثل <a> و <span> ...إلخ.

تسمح CSS بتغيير نوع العرض للعناصر من block إلى inline و بالعكس عن طريق إسناد إحدى القيمتين block و inline إلى الخاصية display.

الشفافية opacity

تتيح CSS لمطور الويب التحكم بدرجة شفافية العناصر عن طريق الخاصة opacity عبر إسناد رقم عشري بين 0.0 و 1.0 إليها، حيث أن الـ٠.٠ تعني حالة عدم الظهور و الـ ١.٠ تعني حالة الظهور التام بشكل طبيعي.

التَّعويم floating

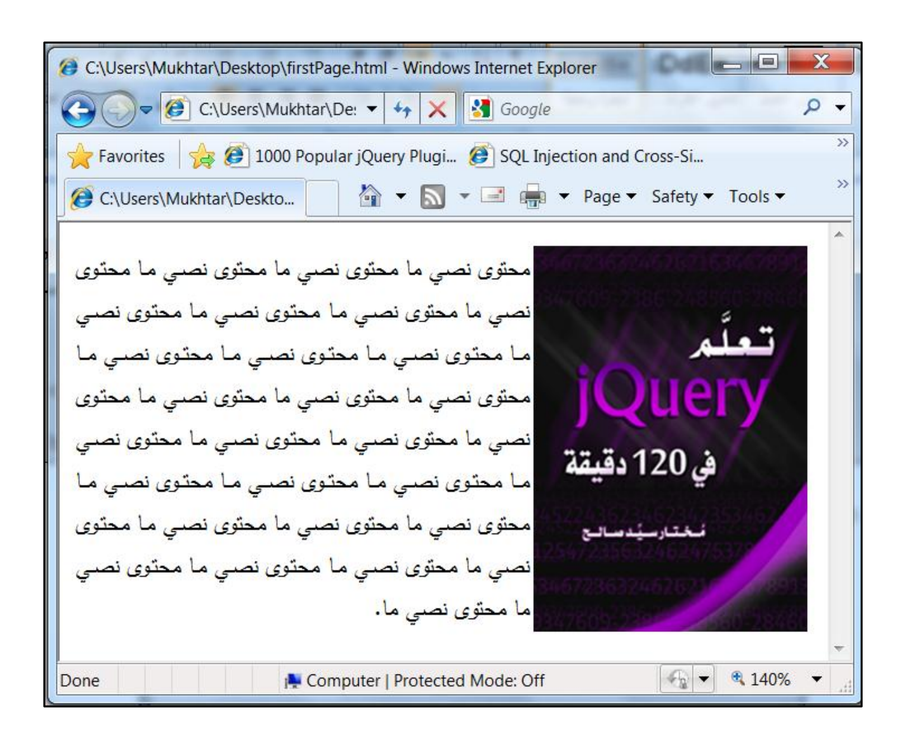
تعتبر خاصية التعويم float من أهم الخصائص التي تقدمها CSS للحفاظ على تموضع متناسق للعناصر حيث يتم تعويم جميع العناصر إلى جهة واحدة بالنسبة لعنصر محدد، و التعويم في CSS تعويم أفقي إما إلى اليمين right أو إلى اليسار left.

عند تعويم عنصر ما فإن العناصر التي تليه سوف تلتف حوله (تعوم) بما يتلاءم و تعويمه، أما العناصر التي قبله فلن تتأثر بالتعويم، لنشاهد المثال التالي:

```
<html>
  <head>
    <style type="text/css">
      p { direction:rtl; text-align:justify; }
```

```
      img { float:right; }
    </style>
  </head>
  <body>
    <p>
       <img src="picture1.gif" />
محتوى نصي ما  محتوى نصي ما محتوى نصي ما محتوى نصي ما محتوى
نصي ما محتوى نصي ما محتوى نصي ما محتوى نصي ما محتوى نصي ما
 محتوى نصي ما محتوى نصي ما محتوى نصي ما محتوى نصي ما محتوى
نصي ما محتوى نصي ما محتوى نصي ما محتوى نصي ما محتوى نصي ما
محتوى نصي ما.
    </p>
  </body>
</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



الشكل ٤٦ : مثال على استخدام خاصية التعويم float

كما تلاحظ فقد تم تعويم الصورة إلى اليمين من النص و التفّ المحتوى الذي يليها بما يتلاءم و موضع الصورة الجديد بعد التعويم، و التفاف النص بهذا الشكل المتناسق لم يكن ليحدث لو تم وضع الصورة في هذا الموضع بطريقة أخرى.

فائدة: جرِّب تغيير حجم نافذة المستعرض و لاحظ احتفاظ الصفحة بتناسقها، ثمّ جرِّب حذف خاصية التعويم من الصفحة و لاحظ الفرق في حال تكرار ذات العملية.

بالطبع و كما ذكرنا سابقاً أصبح واضحاً الآن أن الخاصية float تقوم بتعويم العنصر إلى جهة اليمين أو إلى جهة اليسار، ثمَّ تقوم بتدوير العناصر التالية له لتتلاءم و موضع العنصر المُعَوَّم، و بالطبع يمكن أن يتلو/يسبق العنصر المُعَوَّم الحالي أي قدر من العناصر المعوَّمَة الأخرى و لهذا تتيح CSS منع وجود عناصر عائمة بجوار العنصر الحالي باستخدام الخاصية clear مع العناصر التي لا نرغب بوجود عناصر عائمة بجوارها، حيث يتم إسناد إحدى القيم left لمنع وجود

عناصر عائمة من اليسار أو right لمنع وجود عناصر عائمة من اليمين أو both لمنع وجود عناصر عائمة من الجهتين.

تعليقات CSS

يتم كتابة التعليقات البرمجية في أوراق الأنماط الانسيابيَّة CSS بين */ و /* كمايلي:

```
/*  comment   */
```

أنواع الوسائط Media types

تتيح CSS تخصيص مظهر و شكل الصفحة بناءً على ما يقوم بعرضها، فيمكن منح الصفحة مظهراً و شكلاً معيناً عند استعراضها في مستعرض الويب و منحها مظهراً و شكلاً مختلفاً تماماً عند عرضها على الورق (طباعتها)...إلخ.

يتم تحقيق ما سبق ذكره من خلال وصف نوع media ما قبل مجموعة قواعد CSS، و نوع media آخر قبل مجموعة قواعد CSS أخرى حيث يتم تطبيق كل مجموعة عند الحالة المذكورة في قاعد الـ media@، لنشاهد المثال التالي:

```
<html>
  <head>
    <style type="text/css">

      @media screen
      {
        p{
            text-align:justfy
            direction:rtl;
```

```
            color:white;

         }

        body {

            background-color:green;

         }

      }


     @media print

     {

        p

        {

           text-align:justfy;

           direction:rtl;

           color:white;}

     }

    </style>

  </head>



  <body>

    <p>

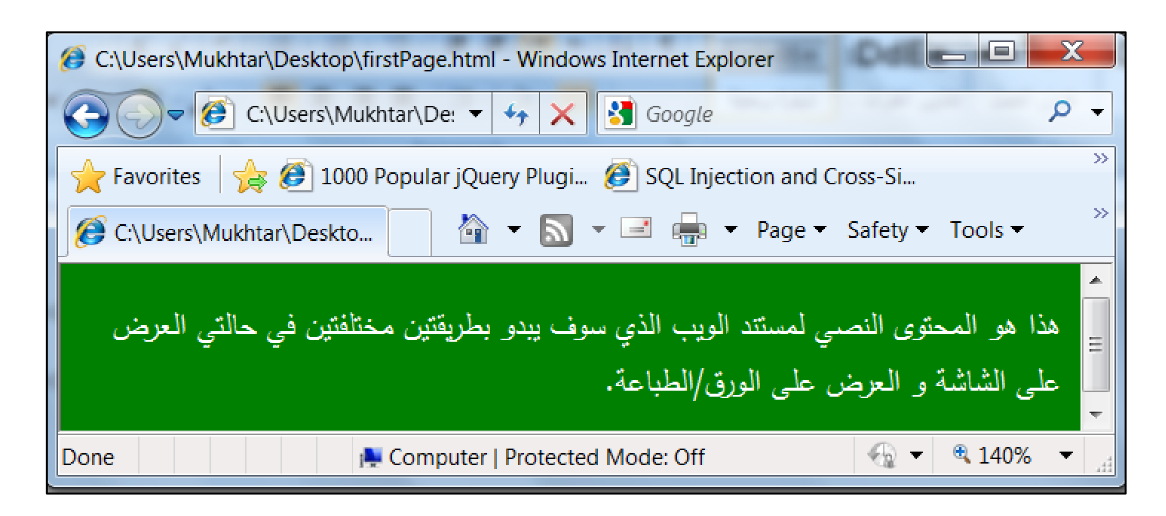
هذا هو المحتوى النصي لمستند الويب الذي سوف يبدو بطريقتين مختلفتين في حالتي العرض على الشاشة و العرض على الورق/الطباعة.

    </p>

  </body>

</html>
```

يبدو المثال السابق عند العرض في مستعرض الويب كمايلي:



الشكل ٤٧ : استخدام القاعدة @media

و يبدو على الورق (في نافذة المعاينة قبل الطباعة في الحقيقة) كمايلي:

Print Preview

1 Page View | Custom | 250

هذا هو المحتوى النصي لمستند الويب الذي سوف يبدو بطريقتين مختلفتين في حالتي العرض على الشاشة و العرض على الورق/الطباعة.

Page 1 of 1

الشكل ٤٨ : استخدام القاعدة @media

يمكن أن نضع بعد media@ إحدى القيم التالية:

القيمة	الشرح
all	جميع أجهزة العرض
aural	للأجهزة الناطقة
braille	الأجهزة المعتمدة على اللمس
embossed	طابعات الأجهزة المعتمدة على اللمس
handheld	للأجهزة الكفيَّة
print	لكل الطابعات
projection	لأجهزة الإسقاط (الراشق مثلاً)
screen	لشاشات الحواسيب
tty	لكل الأجهزة التي تستخدم شاشات ذات عرض محرف ثابت (كالمبرقة الكاتبة).
tv	لكل الأجهزة الشبيهة بالتلفاز

الجدول ١١ : قيم القاعدة media@

و بانتهاء حديثنا عن قواعد الـ Media نكون قد أنهينا –بفضل الله– مناقشة CSS بأغلب خصائصها، و سنتابع في الصفحات التالية استعراض خصائص CSS العصريَّة و التي تعرف باسم CSS3، أمَّا الآن فأرى أن نعرض جدولاً بمُحَدِّدات CSS التقليديَّة ليكون مرجعاً سريعاً عند الحاجة.

دليل محددات CSS

المحدد	مثال	شرح المثال
.class	.Mukhtar	كل العناصر ذات الواصفة "class="Mukhtar
#id	#name	كل العناصر ذات الواصفة "id="name

*	*	كل العناصر
tag	p	كل عناصر الوسم <p>
tag,tag	a,p	كل عناصر الوسم <a> و كل عناصر الوسم <p>
tag tag	div a	كل عناصر الوسم <a> الموجودة داخل <div>
tag>tag	div>a	كل عناصر الوسم <a> الموجودة داخل <div> بشكل مباشر
tag+tag	div+p	كل عناصر الوسم <p> الموجودة مباشرةً بعد عنصر <div>
[attribute]	[src]	كل العناصر التي تحوي الواصفة src
[attribute=v]	[src=image.jpg]	كل العناصر التي تحوي الواصفة src مسنداً إليها القيمة image.jpg
[attribute~=v]	[target~=_blank]	كل العناصر التي تحوي الواصفة target مسنداً إليها قيمة لا تساوي _blank
[attribute|=v]	[href|=http://]	كل العناصر التي تحوي الواصفة target مسنداً إليها قيمة تبدأ بـ http://
:link	a:link	كل الروابط التي لم يتم النقر عليها
:visited	a:visited	كل الروابط التي تم النقر عليها
:active	a:active	الرابط الفعال حالياً
:hover	a:hover	الرابط الذي تمر عليه الفأرة الآن
:focus	input:focus	كل عناصر الإدخال الفعالة حالياً
:first–letter	p:first–letter	أول محرف من كل عنصر <p>
:first–line	p:first–line	أول سطر من كل عنصر <p>

:first–child	p:first–child	كل عنصر <p> يكون الابن الأوّل لعنصره الأب
:before	p:before	إدراج محتوى قبل كل عنصر <p>
:after	p:after	إدراج محتوى بعد كل عنصر <p>
:lang(language)	p:lang(en)	تحديد كل عنصر <p> تبدأ قيمة واصفته lang بالقيمة en

الجدول ١٢ : محددات CSS

CSS 3

مع تطور مفهوم الويب (خصوصاً مع ظهور الـ Web 2.0) ظهرت الحاجة إلى تحسين أوراق الأنماط الانسيابية لتحقيق تصاميم أجمل في مواقع الويب دون الحاجة إلى الإفراط في استخدام برامج التصميم و هذا ما دفع إلى ظهور الإصدار الجديد من أوراق الأنماط الانسيابيَّة و المعروف بـ CSS3.

تضيف CSS3 مجموعة من المحددات و الخصائص الجديدة إلى CSS التقليدية و اصطُلِحَ على تسمية كل مجموعة من هذه الخصائص أو المحددات باسم الوحدة Module، و لعل أهم الوحدات المضافة هي:

- وحدة المحددات الجديدة
- وحدة الحدود و الخلفيات الجديدة
- وحدة تحسين نموذج الصندوق
- وحدة مؤثرات النصوص الجديدة
- وحدة التحويلات الهندسية ذات البعدين
- وحدة التحويلات الهندسية ذات الثلاثة أبعاد
- وحدة الحركات
- وحدة الصفحات متعددة الأعمدة
- وحدة واجهة المستخدم

المستعرضات الداعمة

بالنسبة لـ CSS3 فهي مدعومة من أغلب مستعرضات الويب الحديثة و بشكل أوسع بكثير من دعم ذات المستعرضات لـ HTML5.

الحدود الجديدة CSS3 Borders

تتيح CSS3 إنشاء الحدود ذات الزوايا غير القائمة (Round Corner Borders) بسهولة و يسر و دون الحاجة لاستخدام أي من برامج التصميم (مثل Photoshop) التي طالما استخدمت لتحقيق هذه المهمة عبر خاصية جديدة هي الخاصية border-radius و ذلك بإسناد قيمة إليها بإحدى واحدات القياس، تعبر القيمة المسندة عن نصف قطر قوس الدائرة المقتطع كزاوية للإطار، لنشاهد المثال التالي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

  <head>

    <style type="text/css">

      div

      {

        border:2px solid #a1a1a1;

        padding:10px 40px;

        background:#dddddd;

        width:300px;

        border-radius:25px;

        -moz-border-radius:25px; /* Firefox 3.6 and earlier */

     }

    </style>

  </head>

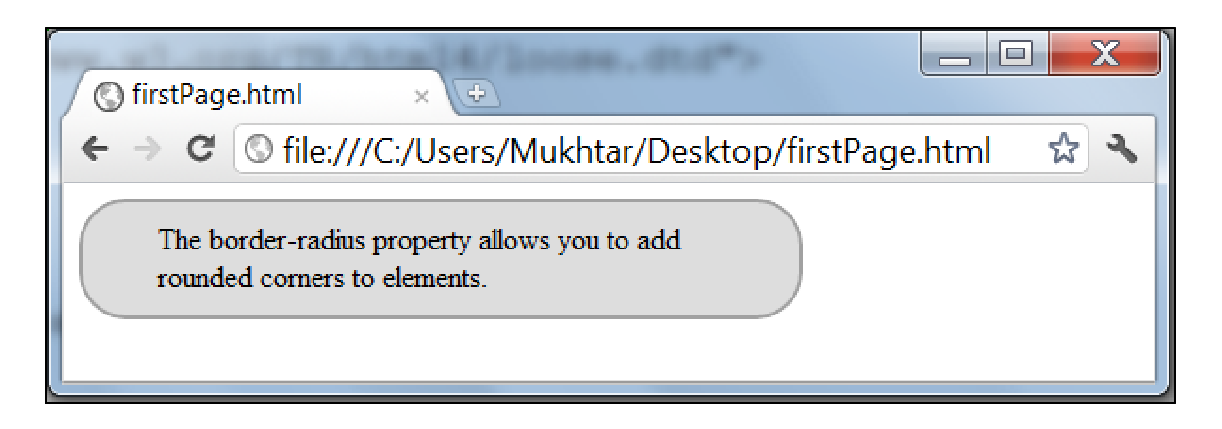
  <body>
```

```
  <div>The border-radius property allows you to add rounded
corners to elements.</div>

  </body>

</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



الشكل ٤٩ : استخدام خاصية border-radius في مستعرض كروم

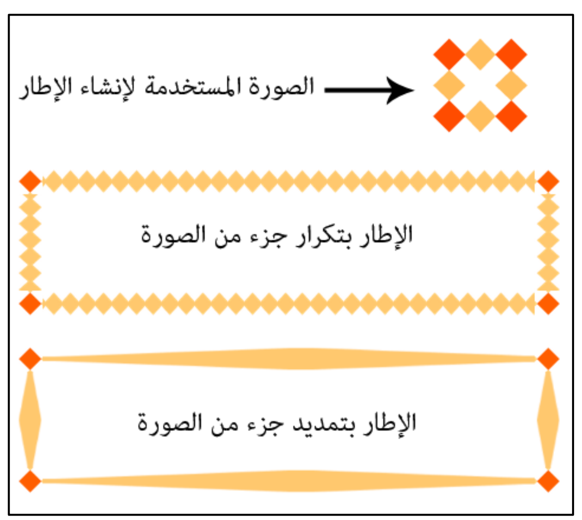
ملحوظة: مستعرض Firefox يعتبر أن اسم الخاصية هو -moz-border-radius

كما تتيح CSS3 إنشاء حدود العناصر بالاعتماد على صورة صغيرة تعبر عن أصغر شكل ممكن للإطار

و يتم إنشاء الإطار بطريقتين:

- repeated: بتكرار جزء من الصورة.
- stretch: بتمديد جزء من الصورة.

و الشكل الجانبي يوضّح الفرق بين الطريقتين المذكورتين.



الشكل ٥٠ : الإطارات الصورية في CSS3

يتم إنشاء الحدود الصورية من خلال تمرير

مسار الصورة إلى الخاصية border–image–source، ثم إسناد إحدى القيمتين repeat أو stretch إلى الخاصية border–image–repeat، ثم إسناد عرض الإطار بالبكسل إلى الخاصية border–image–width، و إسناد قيم مناسبة بالبكسل إلى الخاصيتين border–image–slice و border–image–outset لتحديد جزء الصورة الذي سيتم تكراره أو تمديده لصنع الإطار.

ملحوظة: يمكن استخدام الخاصية المختصرة border–image و التي تجمع جميع الخصائص السابقة بالصيغة border–image:source slice width outside repeat;

لنشاهد المثال التالي الذي يعتمد على الصورة التي عُرِضَت سابقاً لصنع الإطار:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

  <head>

    <style type="text/css">

      div

      {

        border-width:15px;

        width:250px;

        padding:10px 20px;

        -webkit-border-image:url(border.png) 30 30 stretch;

      }

    </style>

  </head>

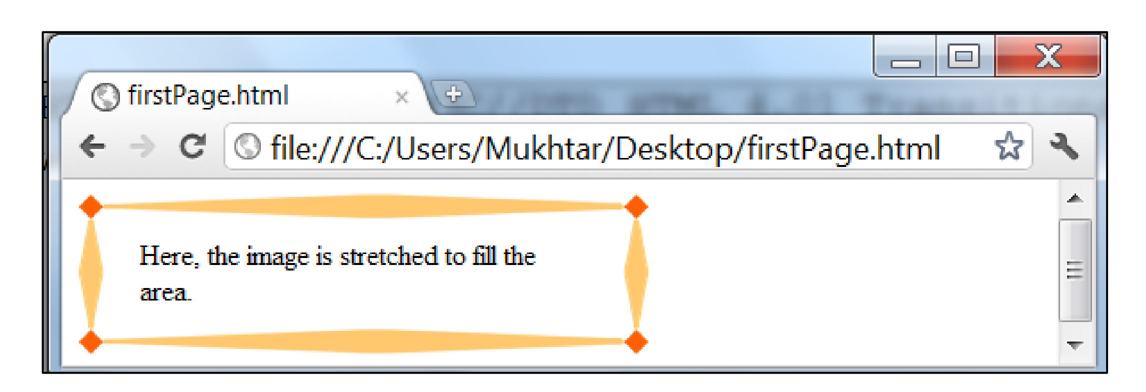
  <body>

    <div> Here, the image is stretched to fill the area.</div>
```

```
  </body>

</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



الشكل ٥١ : الإطار الصوري في مستعرض جوجل كروم

ملحوظة: مستعرض Firefox يستخدم البدائة -moz- قبل أسماء الخصائص المذكورة أعلاه، و مستعرضا Safari و Chrome يستخدمان البادئة -webkit- و مستعرض Opera يستخدم البادئة -o-

تتيح CSS3 أيضاً إنشاء ظل لعنصر ما عبر الخاصية box-shadow و التي يسند إليها قيمة لونية تعبر عن لون الظل و قيم بالبكسل تعبر عن إحداثيات الظل، لنشاهد المثال التالي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

  <head>

    <style type="text/css">

      div

      {

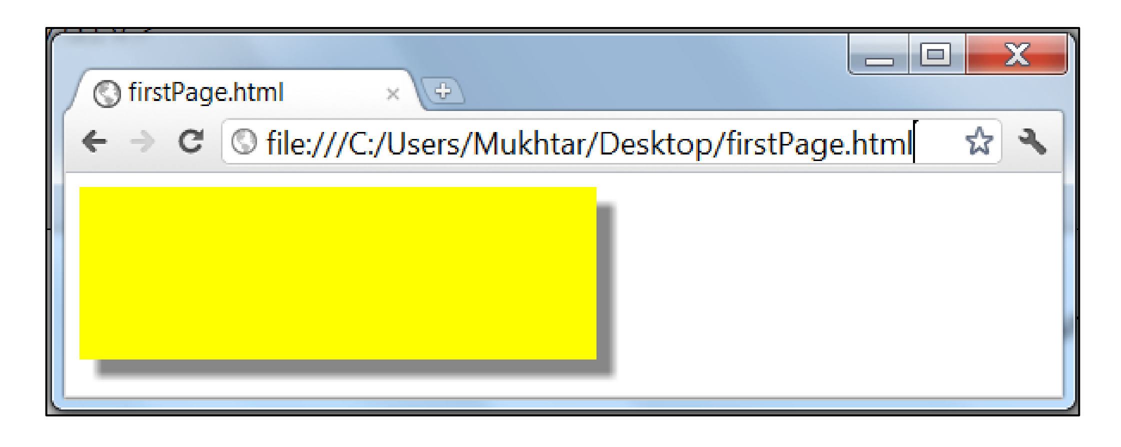
        width:300px;

        height:100px;

        background-color:yellow;
```

```
        -webkit-box-shadow: 10px 10px 5px #888888;
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



الشكل ٥٢ : خاصية الظل كما تبدو في مستعرض كروم

ملحوظة: يستخدم مستعرضا Safari و Chrome البادئة -webkit- قبل اسم الخاصية

الخلفيات الجديدة CSS3 Backgrounds

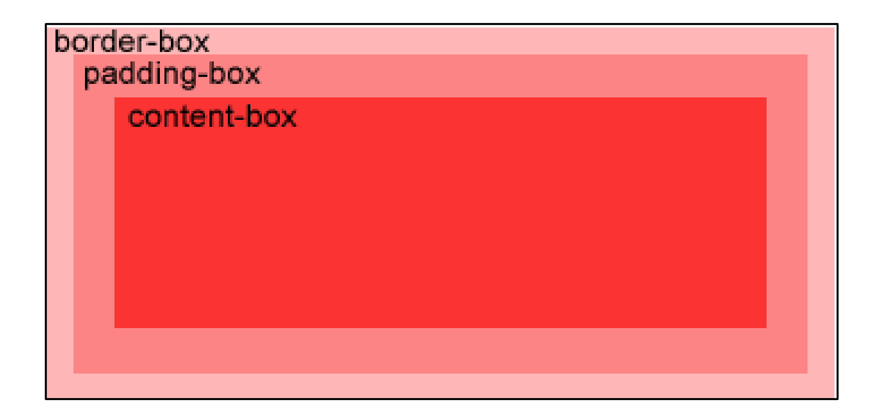
في CSS3 تم إضافة خاصيتين جديدتين لمساعدة مطور الويب على التحكم بالخلفيات بشكل أفضل، الأولى هي الخاصية background-size و التي تسمح بتحديد أبعاد الصورة الخلفية بالبكسل مما يسمح بإعطاء الصورة كخلفية أبعاداً تختلف عن الأبعاد الأصلية للصورة و هو ما لم يكن ممكناً قبل CSS3، يتم استخدام هذه الخاصية كمايلي:

```
background-size: 80px 60px;
```

يعني الرقم الأول إلى عرض الصورة في حين أن الثاني يشير إلى ارتفاعها.

ملحوظة: في مستعرض Firefox تسبق الخاصية بالبادئة -moz-

أما الخاصية الثانية فهي الخاصية background-origin و التي تحدد أحد أجزاء نموذج الصندوق ليتم تطبيق الخلفية على أساسه و يمكن أن تأخذ إحدى القيم context-box لتطبيق الخلفية على المستطيل المحدد بالمحتوى أو padding-box لتطبيق الخلفية على المستطيل المحدد بالمحتوى و الإزاحة الداخلية معاً أو border-box لتطبيق الخلفية على المستطيل المحدد بالمحتوى و بالإزاحة الداخلية و بالحدود و الشكل التالي يوضح الفرق بين المستطيلات الثلاثة.



الشكل ٥٣ : نموذج الصندوق بالنسبة لخاصية الخلفية في CSS3

ملحوظة: في مستعرضي Safari و Chrome يتم استخدام البادئة -webkit-

من التحسينات التي أضيفت إلى خاصية الخلفية في CSS3 السماح بإضافة أكثر من صورة في نفس الوقت كخلفية لنفس العنصر و ذلك عبر الفصل بين كل خلفية و الأخرى بالفاصلة بالشكل:

```
background-image:url(back1.gif),url(back2.png);
```

خصائص تنسيق النصوص الجديدة في CSS3

تم إضافة مجموعة من الخصائص الجديدة لتنسيق النصوص في CSS3 نعرض بعضها في الجدول التالي:

الخاصية	القيم الممكنة	الشرح
hanging-punctuation	none أو first أو last أو allow-end أو force-end	تحدد وضع علامة الترقيم خارج حدود العنصر <p>
punctuation-trim	start أو end أو allow-end أو adjacent أو none	تزيل علامة الترقيم الافتتاحية من بداية كل سطر من أسطر <p>
text-justify	auto أو inter-word أو inter-ideograph أو inter-cluster أو distribute أو kashida أو trim	تحدد طريقة تحقيق المحاذاة عندما تكون قيمة الخاصية text-align مساوية لـ Justify
text-outline	قيمة بالبكسل تعبر عن العرض و قيمة لونية تعبر عن اللون	تحدد عرض و لون الحدود الخارجية للنص
text-overflow	clip أو ellipsis أو string	تحدد ما الذي يتم عمله عندما يصبح طول النص أطول من العنصر الحاوي له

text–shadow	قيمتين رقمتين و قيمة لونية	تضيف ظلاً للنص
text–wrap	none أو normal أو unrestricted أو suppress	تحدد طريقة تحقيق التفاف النصوص
word–break	normal أو break–all أو hyphenate	تحديد طريقة تقسيم الكلمات الطويلة
word–wrap	normal أو word–break	تقوم بتحقيق التفاف النصوص عبر تقسيم الكلمات الطويلة بناءً على الطريقة المحددة في الخاصة word–break

الجدول ١٣ : خصائص تنسيق النصوص الجديدة في CSS3

يعرض المثال التالي استخدام إحدى هذه الخصائص:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

  <head>

    <style type="text/css">

      h1

      {

        text-shadow: 5px 5px 5px #FF0000;

      }

    </style>

  </head>

  <body>
```

```
    <h1>النص المظلل</h1>

  </body>

</html>
```

خصائص الخطوط الجديدة في CSS3

في CSS التقليدية كان من واجب مطوري صفحات الويب استخدام الخطوط الشائعة و الموجودة في أغلب حواسيب زوار الصفحة و ذلك أن المستعرض لم يكن قادراً على عرض أي خط غير موجود في خطوط حاسوب زائر الموقع، مع CSS3 أصبح بالإمكان استخدام أي خط ضمن صفحة الويب بعد وضع ملف الخط ضمن ملفات الموقع و و إسناد مساره إلى الخاصية src لقاعدة @font-face الجديدة كلياً و التي تستخدم كمايلي:

أولاً- يتم تعريف اسم عائلة جديدة باستخدام القاعدة @font-face.

ثانياً- يسند اسم العائلة الذي أُنْشِئَ سابقاً إلى الخاصية font-family بشكل تقليدي.

ملحوظة: مستعرض Internet Exploer يدعم الخطوط ذات النوع eot. فقط بينما تدعم باقي المستعرضات النوعين ttf. و otf.

لنشاهد المثال التالي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

  <head>

    <style type="text/css">

      @font-face

      {

        font-family: myFont;
```

```
        src: url('Sansation_Light.ttf')
        ,url('Sansation_Light.eot') format("opentype"); /* IE */
      }
      div
      {
        font-family:myFont;
      }
    </style>
  </head>
  <body>
    <div>
       يـمكـن اسـتـخدام أي نـوع خطـوط لـهـذا الـنص
    </div>
  </body>
</html>
```

التحويلات الهندسية ذات البعدين 2D Transformation

تمكننا CSS3 من إجراء تحويلات هندسية ثنائية البعد على عناصر الصفحة كالانسحاب translate و التدوير rotate و تغيير الحجم scale و غيرها بإسناد أحد توابع التحويل إلى الخاصة transform.

ملحوظة: مستعرض Internet Explorer يستخدم البادئة -ms- قبل اسم الخاصية، و مستعرض Firefox يستخدم البادئة -moz- و مستعرض Opera يستخدم البادئة -o- و مستعرضي Chrome و Safari يستخدمان البادئة -webkit-

و الجدول التالي يعرض توابع التحويل الجديدة:

التابع	الشرح
translate(x,y)	ينقل العنصر إلى الإحداثيات الممرة له
translateX(n)	ينقل العنصر على المحور X مسافة n
translate(n)	ينقل العنصر على المحور Y مسافة n
scale(x,y)	يغير بعدي العنصر إلى البعدين الممررين له
scaleX(n)	يغير بعد العنصر X إلى القيمة n
scaleY(n)	يغير بعد العنصر Y إلى القيمة n
rotate(α)	يقوم بتدوير العنصر بزاوية α
skew(Xα,Yα)	يقوم بإجراء تحويل الانحراف وفق الزاويتين Xα و Yα
skewX(α)	يقوم بإجراء تحويل الانحراف على المحور X وفق الزاوية α
skewY(α)	يقوم بإجراء تحويل الانحراف على المحور Y وفق الزاوية α

الجدول ١٤ : توابع التحويل الهندسي ثنائي البعد في CSS3

لنشاهد المثال التالي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

  <head>

    <style type="text/css">

      div

      {

        -webkit-transform:skew(30deg,20deg);

        width:100px;

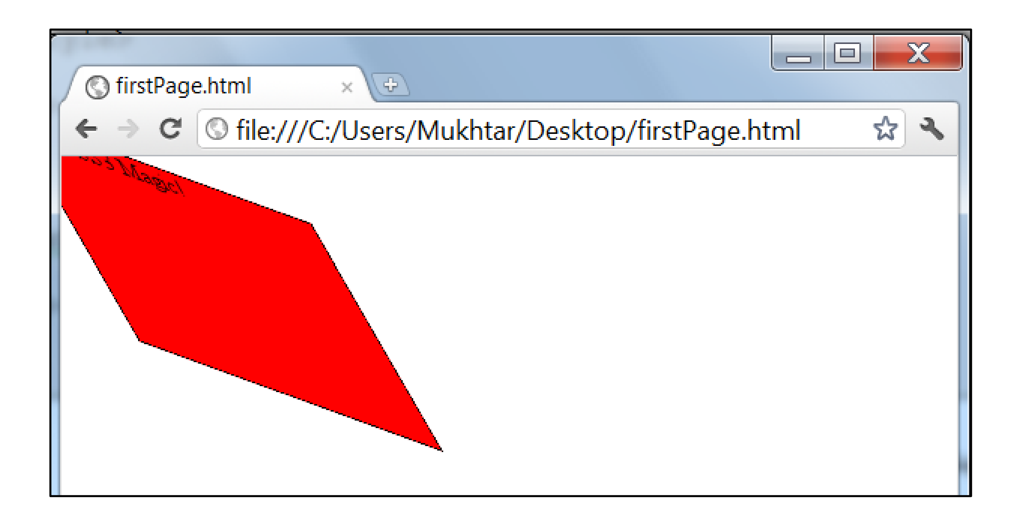
        height:75px;

        background-color:red;

        border:1px solid black;
```

```
        }
      </style>
   </head>
   <body>
      <div>It is CSS3 Magic!</div>
   </body>
</html>
```

و الذي يبدو في مستعرض الويب كمايلي:



الشكل ٥٤ : مثال على تحويل Skew في مستعرض جوجل كروم

ملحوظة: الأحرف deg التي تم تمريرها بعد الأرقام إلى تابع التحويل تعني درجة degree و ذلك تمييزاً عن الراديان و الذي ترمز له الأحرف rad

التحويلات الهندسية ثلاثية الأبعاد 3D transformation

تتيح CSS3 القيام بتحويلات هندسية ثلاثية الأبعاد على العناصر بنفس الأسلوب السابق و لكن باستخدام التوابع التي يعرضها الجدول التالي:

التابع	الشرح
translate3d(x,y,z)	ينقل العنصر إلى الإحداثيات الممرة له
translateX(n)	ينقل العنصر مسافة n على المحور X
translateY(n)	ينقل العنصر مسافة n على المحور Y
translateZ(n)	ينقل العنصر مسافة n على المحور Z
scale3d(x,y,z)	يغير أبعاد العنصر إلى الأبعاد الممرة له
scaleX(n)	يغير بعد العنصر X إلى المقدار n
scaleY(n)	يغير بعد العنصر Y إلى المقدار n
scaleZ(n)	يغير بعد العنصر Z إلى المقدار n
rotate3d(x,y,z,α)	يقوم بتدوير العنصر على المحاور الثلاثة بزاوية مقدارها α
rotateX(α)	يقوم بتدوير العنصر على المحور X بزاوية α
rotateY(α)	يقوم بتدوير العنصر على المحور Y بزاوية α
rotateZ(α)	يقوم بتدوير العنصر على المحور Z بزاوية α
perspective(n)	يحدد منظور الرؤية

الجدول ١٥ : التحويلات الهندسية ثلاثية الأبعاد في CSS3

الانتقال المتحرك Transition

توفر CSS3 طريقة جميلة و مميزة لإنشاء تأثير على عناصر المستند يعرف هذا التأثير الخاص باسم الانتقال المتحرك و الذي يتم تحقيقه عبر الخاصية transition حيث يتم تطبيق قاعدة CSS مختلفة عن القاعدة الأصلية للعنصر خلال زمن محدد مما يظهر العنصر بشكل متحرك.

عن طريق الصيغة:

```
transition: property duration;
```

حيث تعبر property عن اسم الخاصية التي ستتغير و duration تعبر عن الزمن و يمكن تغيير أكثر من خاصية بالصيغة:

```
transition: property1 duration1, property2 duration2,… etc;
```

لنشاهد المثال التالي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

  <head>

    <style type="text/css">

      div

      {

        width:100px;

        height:100px;

        background:red;

        -webkit-transition:width 2s, height 2s;

      }

      div:hover

      {

        width:200px;

        height:200px;

        webkit-transform:rotate(180deg);

      }
```

```
    </style>

  </head>

  <body>

    <div>Hover over me to see the transition effect!</div>

  </body>

</html>
```

ملحوظة: لن أذكر البادئات التي تستخدمها المستعرضات الشهيرة هنا فمن المفترض أنك اعتدت عليها الآن و سأنوه إلى أن الحرف s بعد القيمة الرقمية يعني أن الزمن بالثانية second

حركات CSS3

تتيح CSS3 إنشاء حركات مخصصة في صفحة الويب مما يغني مطور الويب عن استخدام صور الـ gif. المتحركة أو تأثيرات flash و غيرها ضمن الصفحة، و يتم إنشاء الحركات بخطوتين:

أولاً– يتم تعريف اسم خاص للحركة في القاعدة keyframe@ و تحديد ماهيتها عبر تحديد الوضع الابتدائي from و الوضع النهائي to، و بالطبع عند تطبيق الحركة يتم الانتقال من الوضع الابتدائي للعنصر إلى الوضع النهائي خلال فترة زمنية مما يجعلنا نشعر بالحركة.

ثانياً– يتم إسناد اسم الحركة و زمن تنفيذها إلى الخاصية animation الخاصة بالعنصر.

لنشاهد المثال التالي:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
```

```
  <head>
    <style type="text/css">
      div
      {
        width:100px;
        height:100px;
        background:red;
        -webkit-animation:myAnimation 5s;
      }
      @-webkit-keyframes myAnimation
      {
        from {background:red;}
        to {background:yellow;}
      }
    </style>
  </head>
  <body>
    <div></div>
  </body>
</html>
```

المثال السابق يعرض إنشاء حركة بالاعتماد على الحالة البدائية و الحالة النهائية للعنصر أثناء الحركة حيث أن الحالة البدائية تعبر عن شكل العنصر لحظة بدء الحركة و الحالة النهائية تعبر عن شكل العنصر لحظة نهاية الحركة.

يمكن تخصيص شكل العنصر في كل لحظة من لحظات الحركة عبر إنشاء قاعدة CSS خاصة بهذه اللحظة ضمن قاعدة keyframes@ بالشكل:

```
@-webkit-keyframes myAnimation

{

  0% {background:red;}

  15% {backgroung:green;width:150px;color:white;}

  50% {height:350px;}

  100% {background:yellow;}

}
```

حيث أن 0% تعني لحظة بداية الحركة و 100% تعني لحظة نهايتها و القيم الأخرى تعبر عن فترات أثناء حدوث الحركة.

ملحوظة: تحدث هذه التغيرات أثناء الحركة و يعود العنصر إلى شكله الطبيعي عند انتهاء الحركة

الأعمدة المتعددة في CSS3

تسمح CSS3 بعرض محتوى العناصر على شكل أعمدة متعددة Multi–Columns (على مبدأ الجرائد) بطريقتين:

الأولى– عدد أعمدة ثابت و عرض عمود متغير: و يتم ذلك من خلال إسناد عدد الأعمدة إلى الخاصية column–count التي تقسّم المحتوى إلى العدد المذكور من الأعمدة، و يتغير عرض الأعمدة مع تغير حجم نافذة المستعرض.

الثانية– عرض عمود ثابت و عدد أعمدة متغير: و يتم ذلك من خلال عرض العمود الواحد إلى الخاصية column–width و المسافة بين كل عمودين إلى الخاصية column–gap، و في هذه

الطريقة تحافظ الأعمدة على عرض ثابت لكل منها و يتغير عددها مع تغير حجم نافذة المستعرض.

ملحوظة: تستخدم نفس البادئات المذكورة سابقاً مع خصائص الأعمدة المتعددة.

المثال التالي يعرض إنشاء الأعمدة المتعددة بالطريقة الأولى:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

  <head>

    <style type="text/css">

      .newspaper

      {

        -webkit-column-count:3;

      }

    </style>

  </head>

  <body>

    <div class="newspaper">

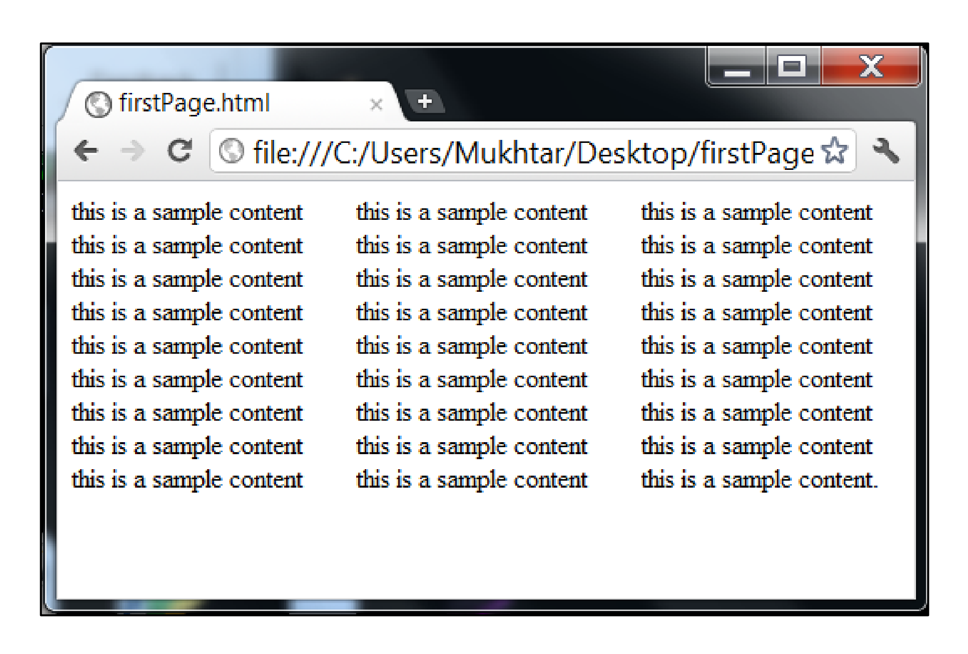
this is a sample content this is a sample content this is a
sample content this is a sample content this is a sample content
this is a sample content this is a sample content this is a
sample content this is a sample content this is a sample content
this is a sample content this is a sample content this is a
sample content this is a sample content this is a sample content
this is a sample content this is a sample content this is a
sample content this is a sample content this is a sample content
this is a sample content this is a sample content this is a
sample content this is a sample content this is a sample content
this is a sample content this is a sample content.
```

```
    </div>

  </body>

</html>
```

يبدو المثال السابق في مستعرض الويب كمايلي:



الشكل ٥٥ : إنشاء محتوى بأعمدة متعددة محددة العدد.

يمكن وضع فواصل ذات مظهر و لون محددين بين الأعمدة عن طريق الخاصية column-rule بشكل مشابه للمثال التالي:

```
column-rule: 4px dotted black;
```

واجهة المستخدم CSS3 User Interface

بالإضافة لما قدمته CSS3 من ميزات لمن ينشئ صفحات الويب فقد قدمت مجموعة من التحسينات للمستخدم النهائي للصفحة عبر الخصائص التي يعرضها الجدول التالي:

الخاصية	القيم الممكنة	الشرح
appearance	normal أو icon أو window أو button أو menu أو field	تحدد لمستعرض الويب كيفية (شكل) عرض العنصر ضمن المستند
icon	auto أو inherit أو رابط لصورة	تتيح جعل عنصر ما بشكل أيقوني
nav-down و **nav-left** و **nav-right** و **nav-up**	auto أو inherit أو معرف أحد العناصر أو اسم إطار	تحدد أين سيتم الانتقال عندما يضغط المستخدم على مفاتيح الأسهم في لوحة المفاتيح
nav-index	عدد صحيح و مميز	يحدد العدد المسند إليها عدد ضغطات مفتاح Tab اللازمة للانتقال إلى العنصر أو بشكل أدق تسلسل الانتقال بين العناصر عند الضغط على مفتاح Tab
resize	none أو horizontal أو vertical أو both	تحدد فيما إذاكان من المسموح للمستخدم تغيير حجم أحد عناصر div بشكل يدوي

الشكل ٥٦ : خصائص واجهة المستخدم الجديدة في CSS3

محددات CSS3

تم إضافة مجموعة من المحددات الجديدة في CSS3 نعرضها في الجدول التالي:

المحدد	مثال	شرح المثال
tag1~tag2	p~ul	كل عنصر قائمة غير مرتبة مسبوق بعنصر فقرة
tag[attribute^=v]	a[href^="https"]	كل عناصر الروابط التي تبدأ قيم واصفتها href بالقيمة https

tag[attribute$=v]	a[href$=".xml"]	كل عناصر الروابط التي تشير إلى ملفات من النوع xml.
tag[attribute*=v]	a[href*="micro"]	كل عناصر الروابط التي تشير إلى عناوين تحوي القيمة micro
:first-of-type	p:first-of-type	تحديد كل عنصر <p> يكون العنصر الابن الأول من نوعه بالنسبة لعنصره الأب
:last-of-type	p:last-of-type	تحديد كل عنصر <p> يكون العنصر الابن الأخير من نوعه بالنسبة لعنصره الأب
:only-of-type	p:only-of-type	تحديد كل عنصر <p> يكون العنصر الابن الوحيد من نوعه بالنسبة لعنصره الأب
:only-child	p:only-child	يحدد كل عنصر <p> يكون الابن الوحيد لعنصره الأب
:nth-child(n)	p:nth-child(2)	يحدد كل عنصر <p> يكون الابن الثاني لعنصره الأب
:nth-last-child(n)	p:nth-last-child(2)	يحدد كل عنصر <p> يكون الابن الثاني لعنصره الأب مع بدء العد من الابن الأخير
:nth-of-type(n)	p:nth-of-type(2)	يحدد كل عنصر <p> يكون الابن الثاني لعنصره الأب
:last-child	p:last-child	يحدد كل عنصر <p> يكون الابن الأخير لعنصره الأب
:root	:root	يحدد العنصر الجذر للمستند
:empty	p:empty	يحدد كل عناصر <p> عديمة

		المحتوى
:target	#news:target	يحدد عنصر #news الفعّال حالياً(الذي تم الوصول إليه عبر النقر على رابط يشير إليه داخلياً)
:enabled	input:enabled	كل عناصر الإدخال المفعلة
:disabled	input:disabled	كل عناصر الإدخال غير المفعّلة
:checked	input:checked	كل عناصر الإدخال المختارة
:not(selector)	:not(p)	كل العناصر عدا عناصر <p>
::selection	::selection	جزء العنصر الذي قام المستخدم بتحديده

الجدول ١٦ : المحددات الجديدة في CSS3

الخاتمة

بهذا نكون قد وصلنا إلى نهاية حديثنا عن CSS3 و بهذا يكون الكتاب قد وفى بوعده بشرح لغتي HTML5 و CSS3 بشكل مكثَّف و بسيط و بهذا تكون أنت قد قطعت ربع المسافة في مشوار تعلم برمجة الويب، و هنا قد تتساءل : و ماذا بعد؟ و في الحقيقة فإن الخطوة التالية (كما أزعم) هي أن تتعلم Java Script أو jQuery ثم تأتي الخطوة المهمة في تعلم إحدى لغات البرمجة من طرف السيرفر مثل PHP أو ASP.NET لتكون قد قطعت ٨٥% من الطريق نحو لقب "مطوِّر ويب Web Developer"، أما الـ ١٥% الباقية فتكمن –برأيي– في عملية نفض الغبار و الأتربة عما تعلمته و الإبقاء على الدر الثمين فقط و تهذيبه عبر القراءة المستمرة و المتابعة الدائمة.

أتمنّى أن يكون كتابي هذا قد ساهم في إعطائك فكرة جيدة عن لغتي HTML5 و CSS3 و أسأل الله أن يكون أسلوبه في عرض الأفكار قد راقك و أرضاك، أمَّا الآن فسأتركك برعاية الله و حفظه و سأدعو لك برحلة هادئة و ممتعة في عالم التطوير للويب، و أنوّه أنّني ما أزال أحبو في عالم الويب و أتعلّم و أرحّب بأي اقتراح أو استفسار على بريدي الإلكتروني mokhtar_ss@hotmail.com و السلام عليكم و رحمة الله و بركاته.

الفهارس

فهرس الأشكال

فهرس الجداول

فهرس المحتويات

عن مؤلِّف الكتاب

(سيرة ذاتية)

<table>
<tr><th></th><th>علميّة</th><th>أدبيّة</th></tr>
<tr><td>المعلومات الشخصيَّة</td><td>• الاسم الصريح : مختار فؤاد سيّد صالح.
• مكان و تاريخ الميلاد : سوريا – البوكمال – ١٩٨٩م.</td><td>الاسم المستعار للأعمال الأدبيَّة : مختار الكمالي.</td></tr>
<tr><td>التحصيل العلمي</td><td colspan="2">• ماجستير في تقانات الويب – الجامعة الافتراضيّة السوريّة – طالب حتّى تاريخه.
• إجازة في هندسة الحاسوب و المعلوماتيَّة بمعدّل جيّد جدّاً – الجامعة السّوريّة الدوليّة – ٢٠١١.
• ثانويّة عامّة – الفرع العلمي – ثانويّة فايز منصور بالبوكمال – ٢٠٠٦.</td></tr>
<tr><td>الخبرات العمليَّة</td><td>• مهندس برمجيّات ويب – بين عامَي ٢٠١٢ و ٢٠١٣ – شركة عجرم للتكنولوجيا – دمشق.
• مهندس برمجيّات – عمل حر – بينَ عامَي ٢٠٠٤ و ٢٠١٠.</td><td>• متفرّغ للعمل الإبداعي منذ عام ٢٠١٣.
• بدأ شاعراً عاميَّاً ثمَّ انتقل للشِّعر الفصيح في بداية ٢٠٠٧.</td></tr>
</table>

<table>
<tr>
<td>من إصداراته</td>
<td>• مشروعُ الفراهيديّ من النظريّة حتّى التطبيق البرمجي – كتاب إلكترونيّ مجّانيّ – دمشق ٢٠١٣.
• HTML5 & CSS3 الخطوة الأولى نحو ويب المستقبل – كتاب إلكترونيّ مجّانيّ – البوكمال – ٢٠١٣.
• تعلّم jQuery في ١٢٠ دقيقة – مكتبة المورد للكمبيوتر – دمشق – ٢٠١٠.
• 3D Game Studio طريقك نحو برمجة الألعاب – ملحق مجّاني مع مجلّة F1 التقنيَّة – دمشق – ٢٠٠٦.
• نُشِرَتْ له مجموعة كبيرة من المقالات التقنيّة في عدد من المجلّات المطبوعة و الإلكترونيّة المتخصصة.</td>
<td>• في غَيَابَةِ الحُبِّ – شِعر – دائرة الثقافة و الإعلام بالشّارقة – ٢٠١٢.</td>
</tr>
<tr>
<td>مهارات العمل البرمجيّة</td>
<td colspan="2">Programming Languages:
• Desktop: C#, VB.NET, VB6, Java.
• Web:
    o Server-Side: ASP.NET classic and ASP.NET MVC, PHP Classic & PHP MVC using CodeIginiter framework., JSP.</td>
</tr>
</table>

	○ Client–Side: HTML 5, CSS 3, java script & jQuery. ○ Technologies: XML, JSON. • Smart phones: Android programming. Database: SQL Server, MySQL, MS–Access. Misc.: MS–Word, MS–PowerPoint, Photoshop, Maintenance skills	
الجوائز و المشاركات	• مسابقة أفضل برمجة و تصميم موقع ويب بتقنيّة ASP.NET المنظّمة من قبل موقع المبرمجين – vb4arab.com المركز الأوّل عن مشروع موقع مدينة أبوكمال – عام ٢٠٠٩. • مسابقة المعلوماتية المركزية الثالثة عشرة – جامعة البعث – حمص ٢٠٠٩. • المؤتمر العلمي الثاني – الجامعة السورية الدولية للعلوم و التكنولوجيا – دمشق ٢٠٠٨. • أولمبياد المعلوماتيّة السوري الرابع – الجمعية العلمية السورية للمعلوماتية – دمشق ٢٠٠٧. • مسابقة المعلوماتية المركزية الحادية عشرة – جامعة البعث – حمص ٢٠٠٧.	• تصفيات مهرجان أمير الشّعراء – الموسم الخامس – الإمارات العربيّة المتحدّة – ٢٠١٣. • جائزة الشّارقة للإبداع العربيّ – الإمارات العربيّة المتحدّة – المركز الأوّل – ٢٠١١. • تصفيات مهرجان أمير الشّعراء – الموسم الرابع – الإمارات العربيّة المتحدّة – ٢٠١١. • جائزة الإبداع الشِّعريّ للطلبة – الجامعة السورية الدولية – المركز الأوّل – ٢٠١٠. • شارك في عدد كبير من الأمسيات و المهرجانات الشِّعريَّة داخل سوريا.

	• مسابقة المعلوماتية المركزية الثامنة– جامعة تشرين – اللاذقيّة ٢٠٠٥. • الملتقى الخامس للإبداع و الاختراع (ورشة البرمجيّات) – حماة ٢٠٠٥. • الملتقى العربي الثالث للمبرمجين الشباب (ورشة البرمجيات) – حلب ٢٠٠٤ – المركز الثالث عن مشروع نظام مكاتب الحوالات الصوتي ٢. • أولمبياد المعلوماتيّة السوري الأوّل – الجمعية العلمية السورية للمعلوماتية – دمشق ٢٠٠٤ – الميدالية الفضيّة.	
العناوين	• العنوان الحالي : سوريا – دمشق • بريد إلكتروني : mokhtar_ss@hotmail.com	